ATARI EXPLORER

THE OFFICIAL ATARI JOURNAL

Summer 1987
Display Until
August 15

$2.95
$3.95
In Canada

Setting Up Your Home Office

Tutorials:

- Basic For Beginners
- Intro To MIDI
- Programming With GEM

Desktop Publishing Primer

Where To Find Your Nearest Atari BBS

Inside: 21 Software Packages Reviewed

- Publishing Partner
- Aegis Animator
- Regent Word
- ST Writer
- Mark Williams C, v.2
- Classic Strategy Games
- MicroLeague Baseball
- ST Alpha-Bytes
- Harrier Strike Mission
- Gato
- Techmate

Hardware Reviews:

- Color Computereyes
- Okidata Microline 292

Bulk Rate
U.S. Postage
PAID
Permit No. 7
Easton, PA 18042

WHY SETTLE FOR 2ND PLACE WITH BASEBALL GAMES?

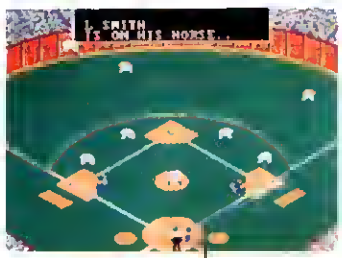
HIT A GRAND SLAM!

MICRO LEAGUE Baseball® II

By popular demand, in its *SECOND* great edition, MicroLeague Baseball is a winner again! And it's so good, it's the MVP (Most Valuable Program) of baseball simulation games.

New levels of baseball strategy. Now you can enjoy an all new level of baseball strategy. Because MicroLeague Baseball II features everything the award-winning first edition had, including: dynamic, high-quality graphics ■ real players and stats ■ true-to-life performance ■ powerhouse teams from the past and present ■ and exclusive endorsement by Major League Baseball and the Players Association. Whether you play a game or a season, if you're looking for managerial excitement, you can't lose.

Natural vs. Artificial Turf



We made it even better! MicroLeague Baseball II goes beyond the first edition, with phenomenal new visuals and sophisticated features such as: stadium dimensions, grass and artificial turf factors, box score and stat compiler *built in*, so you can keep track of your players and their season ■ expanded statistical factors if you're a baseball purist ■ injuries, arguments with the ump and even rain delays. Plus a *"Quik-Play Option"*, if you like to get to the bottom of the ninth inning *fast*.

The "Exclusive" strategy simulation of Major League Baseball.

Major League Excitement Year-Round!

MicroLeague Baseball is *not* a joystick game. Your *mind* — not your reflexes — are in control, and that's what keeps you challenged, game after game. The sights, sounds and stats create a level of involvement no other product can approach. Add to this MLB Season disks and the unique General Manager/Owner disk for drafting & trading players — even creating *your* own fantasy dream teams, and you've got a League Champ!

OFFICIAL LICENSEE MAJOR LEAGUE BASEBALL®

MLB

Official Licensee Major League Baseball, Major League Players Association

MicroLeague Baseball II is available for the Atari ST systems, and will be coming for the Amiga and MAC this summer.

Available at computer and software stores everywhere; or for direct VISA/MC orders simply call **1-800-PLAY8AL**.

MicroLeague I is still available for Apple II Series; Atari 65/130 XE, 800/XL; Commodore 64/128; and IBM PC, PC Jr. and compatibles.

MicroLeague Sports
2201 Drummond Plaza
Newark, DE 19711
(302) 368-9990

T H E **S T** C O M P U T E R S

With the introduction of the ST™ computers comes a new kind of computer language:

PRAISE.

"The most advanced, most powerful microcomputer your money can buy."
—Creative Computing

"The best hardware value of the year."
—Infoworld

"We have spent the last three months evaluating the Atari and have come to the conclusion that it can't be beat as a low-cost telecommunications terminal, drafting workstation, or for quick graphics visualization." *—Microtimes*

"We are most impressed with the clarity of the graphics, with the speed of the disk I/O (input/output), and with the ST's value." ***—Byte Magazine***

"With the impressive ST, Atari has delivered on its promise of power without the price." *—Family Computing*

"Faster and with better graphics capabilities than an IBM®/AT™, it could be a great vehicle for low-cost networks, desktop publishing and visual database management software." ***—Microtimes***

"All of the displays are clear, sharp, readable, and flicker free. We were particularly impressed by the clarity of the high-resolution monochrome."
—Byte Magazine

"The ST's readily apparent strong point is speed. Compared to the Macintosh™, working with the ST is extraordinary."
—John Dvorak, San Francisco Examiner

"Since the pinouts are standard, it is also possible for various software packages to support an even wider range of output devices—even faster printers and high-end plotters." ***—Microtimes***

"The ST is noticeably faster than the Macintosh, not only because of the faster clock rate but because it has a faster disk drive." *—Personal Computing*

"The ST is an amazing bargain, much more of a computer 'for the rest of us' than Mac ever was." ***—Byte Magazine***

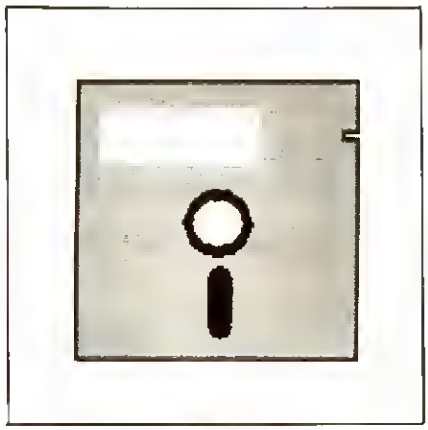
"From here on you had better think of Atari as a major player in the computer game."
—Jerry Pournelle, Infoworld

The 520ST™ with 512K of memory is under $800. The 1040ST™ with a full megabyte is under $1,000. No wonder the experts are impressed.

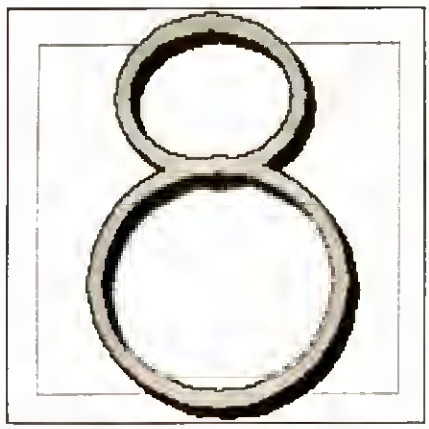
To experience the ST excitement for yourself, see your Atari dealer. For the one nearest you, call 1 800 443 8020. 9:00 AM–5:00 PM Monday–Friday Pacific Time.

ATARI®

Technology So Advanced, It's Affordable.



Eight-bits forever!



Editorial

By JOHN JAINSCHIGG

I recently participated in an open teleconference of magazine editors, sponsored by the Atari 8-bit SIG (Special Interest Group) on CompuServe. Editors from *Atari Explorer*, *Antic*, *Analog*, *Compute*, and other Atari-oriented journals gathered electronically to discuss the future of Atari publishing—in particular, the future of Atari 8-bit coverage.

Some noted a loss of 8-bit reader interest; others complained of a lapse in 8-bit article submissions. Everyone bemoaned the lack of new ideas. How many times could you print a tutorial on redefining character sets or using Player-Missile graphics? Was there anything new left to be said about a five-year-old machine?

As long as I've been involved with microcomputers, I've marveled at the failure of imagination that takes hold just when a piece of hardware is fully understood. Sensing that a frustrated curiosity about specifics is a spur to innovation, the micro industry has placed great stock in postponing comprehension of this kind. Demented documentation, proprietary secrecy, and beaureaucratic ignorance all do their part to delay that frightening moment when all the veils are stripped away.

But when that moment finally arrives, what then? Most innovators, it seems, become disenchanted. Failing to shift their thoughts away from specifics, locked into "let's explore the hardware" mode, they are immediately seduced by the challenges of new equipment. As a result, questions of the commodity value of the old machine—questions that can only be approached when one has mastered all the technicalities—go unexplored.

This disenchantment can be lethal if a machine is unpopular, if its technology is eclipsed by newer and more economical alternatives, or if it is summarily—as is often the case at this juncture—abandoned by its creators. But the Atari 8-bit computers—enormously popular, technically unsurpassed, and firmly supported—are in just the opposite position. No longer on the "leading edge" in one sense, they are on the verge of becoming so in another.

The Atari 8-bits are becoming a kind of commodity—perhaps the first true commodity in the history of personal computing. By Christmas, 1987, another million or more Atari 8-bit computers will be purring away in American homes.

Some will find their first use as entertainment systems—fruit of an enormous resurgence in the popularity of video games. Others will be purchased as utility machines, educational tools, or productivity aids.

They will be bought in toy stores, through discount chains, by mail-order. Only a very few new users will be able to count on the support of a well-established local dealer or to plug in, directly, to the user group network and other information sources. For the most part, they will be *tabula rasa*, their attitudes towards personal computing yet unformed.

How will this market be served? How will it be educated? Will these people have to go through the same agony that we did, thumbing through *De Re*, *Mapping*, and Leventhal at 3:00 a.m., while we look on, disenchanted? Or will we expand our networks to admit them, share our knowledge with them, and by doing so, build better tools for all?

Let's get it together on this one. *Atari Explorer* is actively seeking submissions on topics relating to Atari 8-bit computing. We're looking for useful programs, compact and well- written tutorials, practical feature articles describing how to get the most from your equipment, helpful hints, fillers, and one-liners. Now that we finally know all this stuff, let's put it to work! ■

ATARI EXPLORER

THE OFFICIAL ATARI JOURNAL

SUMMER 1987

FEATURES

REVIEWS

PROGRAMMING

DEPARTMENTS

Cover: Photograph by Jeff MacWright.

Application to mail at second-class postage rates is pending at Netcong, NJ.

POSTMASTER: Send address corrections to Atari Explorer, CN961, Netcong, NJ 07857-0961.

A dozen manufacturers committed to the present and future of Atari computers

Companies To Watch

By DAVID H. AHL, JOHN JAINSCHIGG, and BETSY STAPLES

During our recent travels we have had occasion to make some new acquaintances and renew some old friendships with people who have committed the resources of their companies to development of products for Atari computers. We were pleased to note some exciting, innovative ideas along with a broad base of support in traditional areas. Join us for a quick visit with a dozen companies from whom we're sure you will be seeing a lot more in the future.

Compu-Mates

Charles Faris, a successful record producer and musician with 87 gold and 36 platinum albums to his credit, has turned his considerable talent to designing and producing user-friendly MIDI products at affordable prices. Although Faris's packages are aimed at professional musicians, his retail prices in the $79 to $99 range offer interested amateurs the opportunity to experiment with full-featured, truly professional MIDI products.

Typical of the Compu-Mates line is *K3PO*, an editor for the Kawai K3 synthesizer and Atari ST computer. *K3PO* can save up to 17,800 patches or 5,800 user wave forms on a 3.5″ disk. The wave table editor allows you to "paint" a new waveform with the mouse. The Droid eliminates objectionable harmonics and automatically produces natural sounds that are pleasant to the ear—something that is difficult to do manually.

With the patch editor, the waveform and any sound parameter can be changed in real time, and the sustain pedal allows sustain time to be adjusted rather than remaining fixed.

To get users off to a quick start, an extensive library of sounds is included on the disk. Faris also offers editors for the Casio, Korg, Yamaha, and generic drums synthesizers in versions for both ST and 8-bit Atari computers.

IntelliCreations

Marketing software under the Datasoft name, IntelliCreations has one of the most extensive lines of entertainment software for 8-bit Atari computers. Sam Poole, president of the company, explained that while other manufacturers and retailers were reducing their support of Atari Home Computers, Datasoft was busy preparing new 5.25″ flippy disk packaging with Atari programs on one side and Commodore 64 programs on the other. This approach allows retailers to stock one package that will appeal to two different groups of consumers.

Samuel Poole and Karen Leeds of IntelliCreations

Sam told us that many Atari owners write notes to Datasoft on their warranty cards thanking the company for its continued support of the Atari 8-bit line.

In our November/December 1986 issue, we named several Datasoft products (*Alternate Reality: The City*, *Zorro*, and *Crosscheck*) among the best game releases of 1986, and so far this year, we have reviewed *Mercenary*, *Theatre Europe* and *The NeverEnding Story*. But Datasoft has much more in store for Atari owners this year, including two historic war games (*Bismarck* and *Tobruk*) and three fantasy role-playing games (*Saracen*, *Black Magic*, and *Swords and Sorcery*). We, too, thank them for their support.

Master Designer Software

Master Designer Software designs software under the Cinemaware label, which is distributed by Mindscape. Bob Jacob, president of M.D.S., explained that each Cinemaware package is produced—very much as a movie would be—by a team consisting of a producer, director, screenplay writer, music composer, and computer programmer.

Cinemaware products are noted for their stunning animation—almost movie-like in quality—even on such graphically ungifted computers as the Commodore 64 and IBM PC. Jacob explained that the primary design goal of Cinemaware products is to engross the player completely, much as a good book or movie does.

Furthermore, Jacob, 37, admits to aiming his company's products at middle-aged adults rather than at kids, although he hopes they will be enjoyed by players of all ages.

At the moment, *SDI* is the only Cinemaware product available for the ST, but we are looking forward to *Defender of the Crown*, *Sinbad and the Throne of the Falcon*, and *King of Chicago*.

Aegis Development

Aegis Development, which initially developed its products for the Amiga, has begun to port them over to other computers. Bill Volk, technical vice president of Aegis, explained that this sequence was due simply to the fact that Amiga development systems were the most readily available when the company was formed in mid-1985.

In fact, Bill could hardly contain his enthusiasm for the potential of the newly announced Atari Mega ST and Atari PC and the current outstanding performance and value of the 1040ST, partic-

ATARI EXPLORER
THE OFFICIAL ATARI JOURNAL

Publisher
David H. Ahl

Editor
Elizabeth B. Staples

Senior Technical Editor
John B. Jainschigg

Assistant to the Editors
Julia Winston

Contributing Editors
John J. Anderson
Bob Cockroft
Andy Eddy
Roxanne Farmanfarmaian
Mark Jansen
Arnie Katz
Bill Kunkel
Timothy Onosko
Rick Teverbaugh

Art Director
Peter Kelley

Advertising Sales
Pamela Stockham
(201) 741-5784

Advisory Board
Gershon Blumstein
Richard Frick
Randy Hain
Neil Harris
Larry Samuels
John Skruch
Shiraz Shivji
Leonard Tramiel

Before You Write or Call: Please do not call or write to us about problems with individual Atari products or orders placed with Atari Corp. We will simply forward your query or complaint to Atari Customer Service (P.O. Box 61657, Sunnyvale, CA 94088), where it will arrive days, if not weeks, later than it would have if you had mailed it there yourself. Barbara Hambelton and Diana Goralczyk, Atari's customer service experts, are much better equipped to solve individual problems than we are.

Editorial: Editorial material, including article submissions, press releases, and products for evaluation, should be sent to Atari Explorer, 7 Hilltop Road, Mendham, NJ 07945 (201) 543-6007. Atari Explorer is not responsible for the return of unsolicited manuscripts, tapes, disks, program listings, etc. not submitted with a self-addressed, stamped envelope.

Subscriptions: Domestic: 6 issues, $14.95; 18 issues, $39.95. Canadian: Add $5.00 per 6 issues. Foreign: Add $10.00 per 6 issues. Foreign subscription orders must be accompanied by a check in U.S. funds drawn on a U.S. bank.

Bill Volk, technical VP of Aegis Development

ularly when equipped with the Blitter chip.

The company's latest product for the ST, *Animator ST* combines three animation techniques: *metamorphic*, *cel*, and *color cycle*. The metamorphic technique allows you to create two shapes—say, a pyramid and a star—and have the system draw all the in-between shapes. Cel allows several shapes created with *NeoChrome*—a bird with its wings in three different positions, for example—to be quickly overlaid on one another to create the illusion of motion. Color cycling allows you to cycle repetitively through a range of colors in one image—a waterfall or windmill, for example—to produce the illusion of motion.

Coming soon from Aegis: *Art Pak 1*, a library of animated drawings, including animals, dinosaurs, cars, planes, tanks, and backdrops—the ultimate in clip art.

Aegis also has some fabulous CDI (Compact Disk—Interactive) products on the way . . . watch these pages.

Mission Softs

The story of Mission Softs is a story about a man with a mission. The man: Keith Long; the mission: design an outstanding early learning package for young children. Keith undertook this project about 18 months ago and allows that the learning experience has been unparalleled. "Writing the software," he told us, "was just the tip of the iceberg."

He quickly decided that he did not want an author/publisher arrangement with an established manufacturer but preferred to keep control of the entire development and production process. Today, Keith's face registers a wide variety of emotions as he talks about a printer who delivered 3000 boxes—all flawed—five weeks late, a dealer who grilled him about the product for ten minutes on a long distance phone call and then ordered just one unit, and magazines that demand payment in advance for advertising.

Balancing the frustration, however, is the pleasure of receiving calls and letters from satisfied customers, many of whom, he says, ask if he has any new products yet.

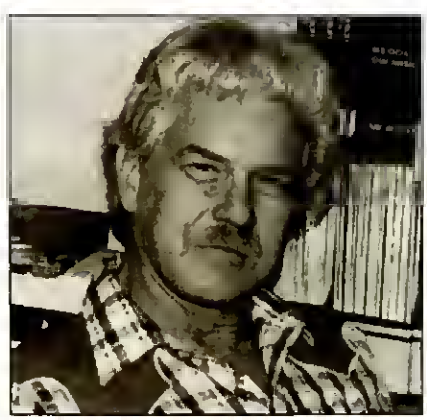
This response is not surprising, considering the thoughtful touches Keith has added to his product, *ST Alpha-Bytes*—a poster to be colored, two sheets of colorful stickers, a spiral bound (lies flat) instruction book, and a classy gold disk label that reads correctly when it is in the drive. See the review of *ST Alpha-Bytes* elsewhere in this issue.

Astra Systems

Astra Systems, is no newcomer to the Atari world, having been a manufacturer of extended performance disk drives for some five years. In fact, we saw an interesting reminder of the company's longevity in the row of Atari 400 computers—the ones with the flat keyboards, remember—being used to run final performance tests on Astra's disk drives.

Drew Featherston, Astra's plant manager, explained that the 400 is the perfect machine for the job; "with its sealed keyboard we can leave it right out on the factory floor without worrying about dust and dirt."

Astra's newest product, the HD+, a combined 20Mb hard disk and 3.5″ floppy drive for the ST, has some very nifty features, including a lighted front panel switch, 36″ cable, small footprint,



Drew Featherston of Astra Systems

versatile backup and restore program, and, perhaps most impressive, a one-year warranty. A review is in the works.

Mastertronic International

Mastertronic International, is a three-year old company originally formed in the U.K. The concept was simple: make a profit by selling a lot of games at the lowest possible prices rather than a few packages at high prices. The original cassette software sold in the U.K. for just £1.99 (about $3.49). The strategy worked, and over the past three years, the company has sold more than 7.5 million units.

A bit over a year ago, Mastertronic brought their low price/high volume approach to the U.S. Today, Mastertronic disk software sells for $9.99, and Stuart Kaye, U.S. vice president, reports that the company has already shipped more than a million units here—an enviable record for the first year of operation.

Stone Age Software

The Atari ST is a perfect general purpose computing system for college and graduate students, particularly engineers and other technical types whose research requires sophisticated graphics and number-crunching capabilities. Brothers Virgil and Steve Mehalek co-founded Stone Age Software to provide Atari ST software tools for science, engineering, mathematics, and related fields of study at prices students could afford.

Stone Age currently offers a suite of three software packages. *Ultracalc* is a desktop calculator with 32-bit precision, which is capable of operating in decimal, simple binary, 1's complement, octal, or hexadecimal modes, and incorporates six memory registers. It sports a wide variety of built-in operations (including bitwise Boolean logic ops!) and an array of useful constants, such as Avogadro's number (6.02×10^{23}, last time we counted). The calculator is configurable either as a stand-alone program or as a desk accessory and is available with algebraic or RPN interfaces (appropriate for Forth programmers and members of Solidarnosc), for only $24.95.

Encrypt is a high-speed binary encryption system that can make a reversibly incomprehensible jumble out of any file at better than a K per second, for $19.95. *Lewis-123* is a "chemistry spreadsheet"—a graphic tool for chemists that assists in the modeling of com-

plex molecules, emphasizing covalent bonding.

Lewis-123 calculates the number of bonds and unshared electrons in a molecule under study, computes formal charge, and performs simple error-checking on bonds as they are formed in the molecular diagram. Extensive on-line help is supported, including full stats on all (non-transition) elements in the periodic table (imagine not having to remember that the second ionization energy of Gallium is 1978.8 Kilo-

Steve Mehalek, co-founder of Stone Age Software.

joules—what a luxury). The program supports output to *Degas* files and uses standard *Degas* printer drivers. It sells for $29.95.

ICD

If you are an 8-bit Atari owner, you should know about ICD, Inc. of Rockford, IL. For the past several years, ICD has been producing quality hardware and systems software products for the Atari line.

Some highlights of ICD's extensive line: R-Time 8, a battery-backed system date/time clock in the form of a piggyback cartridge that lets you run other cartridge-based software in normal fashion. The Printer Connection, a "smart cable" with built-in interface electronics that lets you connect your Atari to any Centronics-compatible parallel printer. And, for the technically-minded: Multi I/O a five-function Input/Output interface that incorporates a 256K (or 1Mb) Ramdisk, parallel printer interface, RS-232 interface, intelligent print spooler, and even a hard disk interface, compatible with SASI or SCSI fixed disks.

Sense Software

Chester Sensenig has been programming down to the bare metal for a long time. A mathematician with an affinity for 68000 assembly language, he started writing productivity applications at the prompting of his son, Douglas, who was studying business management and whose work required innovative software tools.

Upon Doug's graduation, the father-son team founded Sense Software to market Chester's line of Atari ST business products to a wider audience.

Currently, Sense is offering *DBSense*, a text-based, relational database program incorporating its own, easy-to-learn, application programming language (à la *dBase*). *DBSense* applications can access multiple files, creating, for example, a monthly report out of data drawn from separate accounts payable and receivable and income files.

Math (with 16-digit, double precision, appropriate for most financial calculations) functions and operations, and a full text editor are supported in the *DBSense* environment, making it suitable for a wide variety of data management applications, from accounting to mass mailing. At the moment, *DBSense* employs a "command line" interface, but Sense is considering making the program fully GEM-compatible in the near future.

White Lion Software

At a time when talk of desktop publishing and other forms of fancy printing focusses on 16-bit machines, Ira Brickman, president of White Lion Software, is concentrating the development efforts of his company on the tried and true combination of a 48K disk-based 8-bit Atari running *Atariwriter*,

Ira Brickman of White Lion Software.

Atariwriter Plus, or *Paper Clip*.

The Ridge, NY, firm has just released *Qwik Pix* and *Qwik Pix PC* (for *Paper Clip*), programs that allow you to integrate graphics from *Print Shop* or public domain sources into documents created with your word processor.

Qwik Pix co-author Brad McCall lives far from Ridge, NY, in Georgia, and he and Ira have demonstrated their faith in technology by completing the entire programming project via modem; they have never met, but the fruit of their telecommunicative cooperation is available to Atarians across the country. ■

Where To Find The Companies To Watch

Aegis Development
2115 Pico Blvd.
Santa Monica, CA 90405
(213) 392-9972

Astra Systems
2500 S. Fairview, Unit L
Santa Ana, CA 92704
(714) 549-2141

Compu-Mates
8621 Wilshire Blvd., Ste.177
Beverly Hills, CA 90211
(213) 271-7410

ICD
1220 Rock St.
Rockford, IL 61101
(815) 968-2228

IntelliCreations
19808 Nordhoff Pl.
Chatsworth, CA 91311
(818) 886-5922

Master Designer Software
5743 Corsa Ave., Ste. 215
Westlake Village, CA 91316
(818) 889-1537

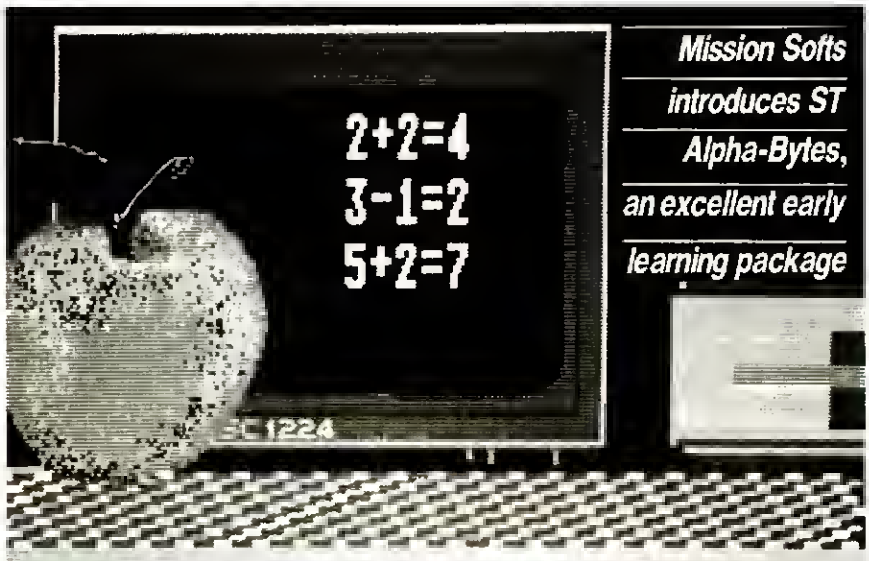
Mastertronic International
711 W. 17th St., Unit G-9
Costa Mesa, CA 92627
(714) 631-1001

Mission Softs
P.O. Box 3916
Seal Beach, CA 90740

Sense Software
105 Iron Mountain Rd.
Warwick, NY 10990
(914) 986-2711

Stone Age Software
P.O. Box 1216
Amherst, NH 03031
(603) 881-7689

White Lion Software
P.O. Box 357
Ridge, NY 11961
(516) 924-6071



Mission Softs introduces ST Alpha-Bytes, an excellent early learning package

Atari Classroom

By BETSY STAPLES

The thing that struck me first about *ST Alpha-Bytes*, before I had even booted the disk, was what I think is best described as the "attitude" of Mission Softs. As you read through the small (4″ × 6″) instruction booklet, you quickly realize that Keith Long, author of the program, really *cares*.

He cares enough to provide an unprotected disk, because he knows that young children will be using it. He cares enough to include in the booklet a "minicourse in child development", because the parent or teacher who is using the program to "provide a setting that is congenial to learning" will be better equipped to do so if he "has an understanding of what happens to a child as he is growing." He cares enough to enclose in each package two sheets of colorful stickers and an alphabet poster to color.

And finally, he cares enough to offer to modify your copy of *ST Alpha-Bytes* if there is some special change that you think will help your child learn better. If he incorporates your suggestion in a future release of the program, he won't even charge you for his time.

Using the Program

The program itself is very simple. The letters of the alphabet are displayed in boxes across the bottom of the screen. When a letter is chosen, either automatically by the program or by the child, a large representation of that letter in both upper and lower case appears on the left-hand side of the screen. A drawing of an object, the name of which starts with the letter displayed, appears on the right, and three words appear in the middle. The child must choose the word that describes the picture.

System: Atari ST
Price: $29.95
Summary: An excellent, carefully-implemented early learning package for ages 2 to 6.
Manufacturer:
Mission Softs
P.O. Box 3916
Seal Beach, CA 90740
(213) 439-6281

The display is drawn in four-color medium resolution, and you have a choice of a black or white background. The colors add a little pizzazz, but it is nice to know that the program runs equally well on a monochrome system.

The child can control the program with either the mouse or the keyboard, but our playtesters preferred to use the keyboard, even though it was more difficult for them to locate specific letters that way.

Each time a letter is chosen, the same three words are displayed, but the picture changes. The first time the letter R is selected, for example, a rocket may be drawn. The next time a ring or a rake will appear.

The words have been carefully chosen to eliminate the possibility of confusion for the child. Each word in a given group uses different phonemes, so a child with even rudimentary reading skills can distinguish one from another just by sounding them out. He does not, for example, have to choose between *tree* and *three*.

The pictures themselves are, for the most part, simple outline drawings that are easy to identify. Our five-year old playtester did have a bit of trouble with a string of eight-legged ants, however; he knew that insects have only six legs each but couldn't find *arachnid* in the word list.

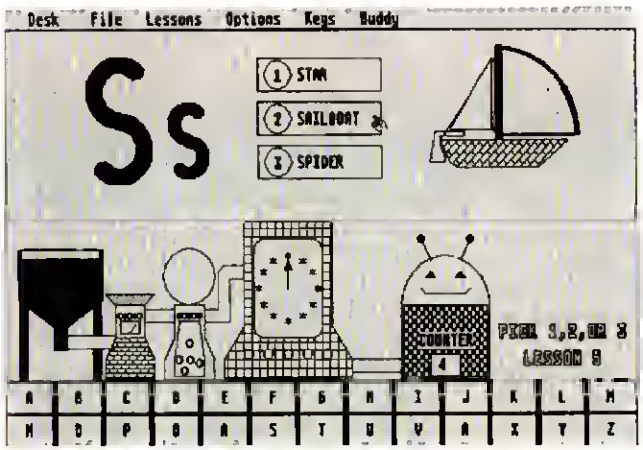
ST Alpha-Bytes offers five lessons or variations on the main theme. Lesson 1 automatically reviews the alphabet in order. Lesson 2 reviews the letters in random order. Neither of these modes requires any input from the child.

Lesson 3 is a quiz mode, which displays the letters in order and asks the child to choose the correct word each time. Lesson 4 does the same in random order. Each correct answer is rewarded with a Happy Face. In the quiz modes, the child's score is maintained on a score sheet, which can later be displayed or printed out.

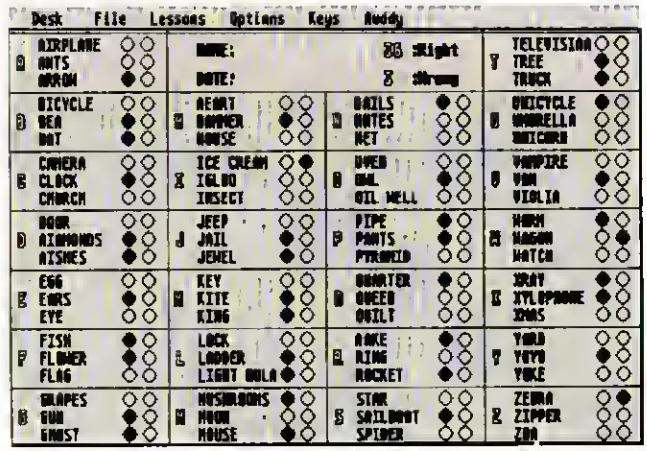
Lesson 5, the one which comes up when you boot the program, is also the one our playtesters enjoyed most. It allows the child to select the letter that will be displayed. If he picks the correct word, the "Counter Machine," located in the middle of the screen, swings into action, and the child can watch and listen as a little ball makes its way from a storage tank through a variety of gizmos across the screen to a smiling, antennaed Counter where its arrival increases his score by one.

ST Alph-Bytes is intended for use by children aged 2 to 6. Those at the low end of that range will require lots of adult assistance. For these very young children, the computer serves primarily to motivate those who might not otherwise be willing to concentrate on learning the letter sounds.

Older children, however—even those over 6—will enjoy playing both by themselves and with adult guidance. Our older playtesters even had fun helping their five-year old brother sound out the words on the screen.



ST Alpha-Bytes main screen.



The score sheet can be printed or displayed.

Documentation

My only complaint about an otherwise carefully and professionally produced program is (gosh, I'm tired of saying this!) the documentation.

The cute little booklet, each of whose pages is decorated with the silhouette of a different animal, covers program operation thoroughly and offers, as mentioned above, a great deal of extra information.

HOWEVER, the writing style is nothing short of appalling. In an effort to sound pedantic while dodging the specter of sexism, author Arlene C. Magel has evolved a stilted and ungrammatical style that should set your teeth on edge.

Sentences like "These improved bodily powers indicates readiness to handle themselves at the family dining table" and "Although four-year-olds may count to ten, his number concept barely goes beyond two or three" make it very difficult to concentrate on the content of the manual.

The good news is that the child using the program is not subjected to the author's errors, and while it may be painful, I assume that an educated adult will be able to separate the message from the medium and escape unscathed.

From the child's perspective, then, which is after all what should concern us, *ST Alph-Bytes* is an excellent program. It provides motivation, entertainment, and education in a skillfully designed package that he will enjoy using in different ways over a period of years. We recommend it. ■

Desktop Publishing For The Atari ST

A bit of background and a review of Publishing Partner from Soft Logik

By TIMOTHY ONOSKO

We live in a world of information. Every day we are bombarded by messages of all kinds: audio, visual, and electronic. But it is paper—the stuff you now hold in your hands—that has withstood the challenge from the other information media and has flourished.

Despite the onslaught of information from radio, television, and new forms of digital communications, the death of the printed word, to paraphrase Mark Twain, has been greatly exaggerated. Ironically, it is the computer, which some thought would eventually eliminate the need for paper (remember the "paperless office"?), that has given new life to the traditional craft of publishing.

"Desktop publishing" is the most recent major trend in personal computing. Essentially, it is the ability to set text in type, mix it with graphics, and finally, compose a printed page—all without the need for manual tools such as scissors, paper, and glue. In the midst of the hoopla and hype that surround the desktop publishing movement are some significant changes in the way ideas are prepared for publication.

Desktop publishing programs for other computers have existed for quite some time, but it wasn't until recently that a full-featured system was available for Atari ST computers. *Publishing Partner* from Soft Logik is the first effort to bring these exciting new capabilities to the Atari machines.

The Word and the Page

A Bit of History

Typography is the illustration of words on paper, an art in itself that has been evolving since the fifteenth century. Most historians identify Johann Gutenberg, a metalsmith, as the inventor of movable type in about 1440, but there is some evidence that the Chinese used similar cast metal type as many as 400 years earlier. Gutenberg's 1455 Bible, however, set the wheels of European publishing in motion by the end of the 15th century.

When Gutenberg's city of Mainz, Germany was sacked in 1462, the center of printing and typography moved to Italy—to Rome and Venice. It was in Venice that a scholar named Aldus Manutius established a press to publish works of Greek literature and the writings of the Renaissance humanists. For his works, Aldus designed the first important Roman-style typefaee and originated italic type, based on the fluid penstroke of hand copyists.

Individual pieces of cast metal type remained the standard for more than 400 years, until 1886 when the first Linotype machine was installed at the New York *Tribune*. It was the first successful attempt to automate typesetting and the brainchild of Ottmar Mergenthaler, a German immigrant who first tinkered with the idea while working in a Baltimore machine shop. The Linotype machine produced the first real mass-communications revolution, a huge wave of inexpensive newspapers, books and magazines.

The Linotype (and variations on it), which cast *slugs* of type from hot metal, was the predominant method of typesetting until phototypesetting machines were introduced after World War II.

Phototypesetting produces type by exposing photographic paper to an image of each individual letter. The image may come from a film negative that is enlarged or reduced, or from a picture of the character on a computer-driven cathode ray tube similar to a television set or video monitor. Today, most professional typesetting is done by phototype machines.

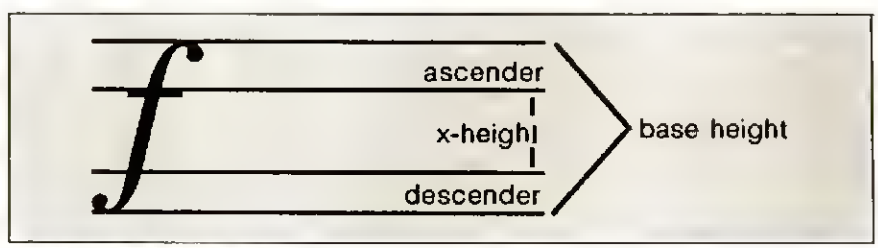


Figure 1. Determining point size.

But the sun is setting on the era of phototypesetting. The laser, that hairthin beam of light, is virtually revolutionizing and reinventing typesetting. Laser typesetters, because they can produce complete, finished pages of type and graphics, are usually referred to as *imagesetters*. While today's laser imagesetters still rely on photographic technology—silver-based photo papers and chemical developing—to produce high-quality type and pictures, the laser printer is on its way to making even these elements obsolete.

Pages are composed of columns of type, each width determined by the lengths of lines. These lengths are measured, not in inches, but in picas. Six picas equal one inch. Wide columns of small type (9 or 10-point) are difficult to read. Narrow columns of larger type (12 or 14-point) produce "rivers" of white space that also reduce legibility.

Figure 2. Wide columns of small type are difficult to read.

Laser printers use a beam of light to create an image on paper in much the same way that common office photocopiers operate. (See "Printing By Laser" elsewhere in this issue.) The resolution of current laser printers is relatively low (most offer "only" 300 dots per linear inch), but good enough to furnish "okay-quality" typesetting and even some coarse photographic reproduction. In the near future, however, the output of laser printers is almost certain to rival that of high-quality phototypesetting and laser imagesetters.

The Basics of Typography

A few words of explanation about some of the basics of typography are in order.

Typewriter and computer-printer characters are measured by their *pitch*. That is, the number of characters that fit in one horizontal inch. (The Courier typeface on a typewriter or daisy wheel printer, for example, comes in 10- and 12-pitch.) Ten-pitch, with 10 characters to the inch, is larger than 12-pitch, which has 12 characters to the inch.

Type, on the other hand, is measured in *points*, 72 of which equal one inch. This is the basic and essential measurement of printing and typesetting.

The point size of type has nothing to do with how many characters fit on a line. The size of a typeface—10-point, 12-point, 18-point, and so on—is determined by measuring the maximum size from the tips of portions protruding downward (*descenders*) to the very tops of segments that extend upward, called, naturally, *ascenders*. Printers also call this the *base height*.

The section of a character exclusive of its ascender or descender is called the *x-height*. Because of the unique proportions of certain typefaces, it is possible for 10-point type in one style (Helvetica, for instance) to be larger than 12-point type in another (such as Times Roman). (See Figure 1.)

While vertical line spacing on typewriters and computer printers is relatively straightforward—single, double, or triple spaces between lines—line spacing is much more flexible in true typesetting. The space between lines, called *leading* (pronounced "ledding" and a holdover from the days when strips of lead were inserted between lines of type), is also measured in points. Tight line spacing is about equal to the size of the type—10-point type with 10-point leading, for example. Loose spacing incorporates more leading. Much of the type you read in newspapers is set slightly tight—about one point greater than the type size, e.g., 9-point type with 10-point leading.

Pages are composed of columns of type, the widths of which are determined by the lengths of the lines that compose them. The length is measured, not in inches, but in *picas*; six picas equal one inch. Wide columns of small type (9- or 10-point) are difficult to read (see Figure 2). Narrow columns of larger type (12- or 14-point) produce "rivers" of white space that also reduce legibility (see Figure 3). When all the words in a column don't end at exactly the same place on the line, it is called *ragged right* composition (see Figure 3). When they all match (as on this page), the column is said to be *justified*.

Justification is tricky and involves letter spacing as well as word spacing to produce an easy-to-read document. Strangely, studies show that most people seldom notice whether a column is justified or not while they are reading.

Lines on a page are called *rules* and, again, are measured in points. One-point rules are most common, but contemporary design has made popular narrower lines of ½ and even ¼ point, called *hairline* rules (see Figure 4). Likewise, the widths of lines that outline boxes and circles are also measured in points.

You may be asking yourself why all of this is important. Typography is a rich and complex art, one worthy of further study. (A good place to start is the books written by type historian and designer Stanley Morrison, which can be found in any good library.) The ground rules of typography have been established over centuries and, while once the sole domain of typographers, the world of picas and points and the elements of page design are becoming more and more important to personal computer users.

Just as we moved from typewriters to personal computers for the preparation of words and ideas, we are moving from simple computer printers into more complex typesetting and graphic arts techniques. While laser printers might seem an expensive option today, they

Pages are composed of columns of type, each width determined by the lengths of lines. These lengths are measured, not in inches, but in picas. Narrow columns of larger type (12 or 14-point) produce "rivers" of white space that also reduce legibility.

Figure 3. Narrow columns of large type suffer from reduced legibility.

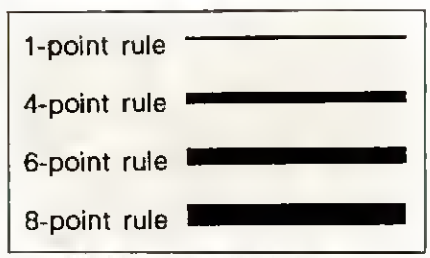


Figure 4. Rules are measured in points.

Publishing Partner

System: Atari ST; 1040, monochrome monitor, and second disk drive recommended.
Price: $149
Summary: Desktop publishing program puts publishing power into the hands of ST users.
Manufacturer:
Soft Logik Corporation
4129 Old Baumgartner
St. Louis, MO 63129

will one day replace most conventional computer printers as certainly as the computer replaced the typewriter.

And these complexities are not going to simplify themselves just because personal computer users have not yet been exposed to them or are unwilling to adapt. The ultimatum: learn now or be prepared to return to pencil and paper.

The Page and the Screen

Introduction to Publishing Partner

Your initiation is now complete, and you are ready to become an apprentice in the field of electronic typesetting and publishing. The best place to start is with a tour of *Publishing Partner*, a typesetting and page composition program that, just five years ago, would have been inconceivable on a personal computer. What makes it possible today is the speed of the 68000 processor and the enormous memory of your Atari ST computer.

A desktop publishing program like *Publishing Partner* accomplishes several different functions. It sets type in columns and allows you to determine the style of each word—plain, bold, italic, bold-italic, etc. It arranges these columns on the page and incorporates graphics—rules and drawings. Finally, it issues commands to a printer—dot matrix or laser—to print the composition on paper. More on this last point later.

Publishing Partner offers an electronic analogy to the paper page on which you can move all these elements by pointing to them with the mouse and dragging them around. The popular term for this kind of analogy is WYSIWYG—"Whizzywig," for What You See Is What You Get. It is not, strictly speaking. In fact, critics have noted that it is, at best, really AWYG—Approximately What You Get. Since *Publishing Partner* can print to a laser printer at 300 dots per inch, true WYSIWYG would mean that even the ST high-resolution monochrome screen would be capable of showing only a two-inch wide swatch of the page.

Nonetheless, the approach of *Publishing Partner* to WYSIWYG is a practical method of putting publishing power into the hands, not just of eager amateurs, but of graphic artists and professionals more accustomed to traditional ways.

A *Publishing Partner* file can contain up to 99 separate pages. Larger documents can be created from separate files, each with its own page numbering system. Page size can be chosen from several stock sizes—letter ($8\frac{1}{2}'' \times 11''$), legal, index card ($3'' \times 5''$), business card, plus European sizes A4 and B4—or can be specified by the user, from 1 to 18 inches square. Orientation can be "portrait" (taller than wide) or "landscape" (wider than tall).

Master pages can be created so that graphic material necessary for a consistent appearance can be automatically included. This is important for continuing elements like page numbers, chapter titles, and other design motifs. Left and right-hand master pages can be designated when you choose the double-sided option upon creating a new document.

For visual measurement, vertical and horizontal rulers (on the left side and top of the page window) can be displayed. Ruler measurements can be shown in units of inches, centimeters, or picas and points. The horizontal ruler also shows tab stops. "Guides" to aid in the placement of text and graphics can be set on both rulers. (The program manual never really makes it clear what tab stops and guides look like or how to set them, but a little experimentation provides the answers.)

Text and pictures are *imported* to the program; text from any ASCII text file, and pictures from graphic painting programs like *Degas* and *Neochrome*. (In addition, the "tiny" compressed graphic file format is also accepted.) The placement of both involves creating regions using an arrow-shaped pointing tool. Alternately, columns for text can be created by choosing the Create Columns option from the menu bar. (This is the preferred method for creating evenly-spaced magazine and book pages. Freehand text columns are probably more useful for posters and advertisements.)

At the right of the screen is a "toolbox" that contains controls for most of the graphic and drawing functions of the program. Without leaving *Publishing Partner*, you can create circles, ellipses, boxes, boxes with rounded corners, irregularly-shaped polygons, and horizontal, vertical, and diagonal lines. Another drawing tool allows freehand drawing.

This is object-oriented drawing, not to be confused with bitmap drawing programs like *Neochrome* and *Degas*. These objects can be resized and repositioned anywhere on the page. You cannot, however, make individual, pixel-wide changes in the shapes you create.

The shapes themselves can be filled with patterns. Thirty-eight patterns are provided and can be edited. Each shape, too, can be drawn in various line widths. Line and rule patterns can also be edited.

Figure 5. Publishing Partner offers a bewildering array of type enhancements.

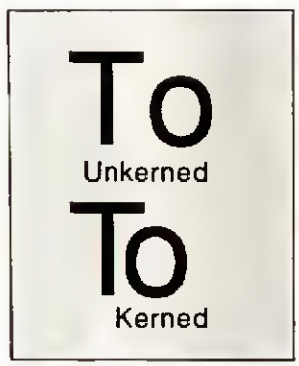


Figure 6. Kerning improves the appearance of type.

Using Text

Text is used in two ways in the design of a printed page—as headlines and as body copy. Creating headlines is simple with *Publishing Partner*. A freehand text block is indicated, then the typeface (or font) and size (3 to 72 points) are selected. Type can then be entered directly from the keyboard. Body copy can either be entered from the keyboard or "poured" into a column by importing an ASCII file. If the text is longer than any single column, it can flow from one column to another by linking the columns, indicating the continuation of one file.

Where *Publishing Partner* really shines is in its ability to manipulate and enhance text. Any line, word, or character within a column, for example, can be any size type. Furthermore, various enhancements or attributes, such as boldface or italic type styles, can be desig-

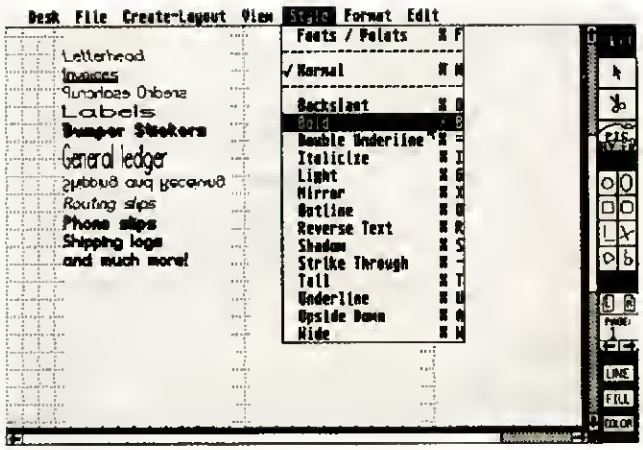


Figure 7a. From the Publishing Partner Style menu a variety of type styles can be selected.

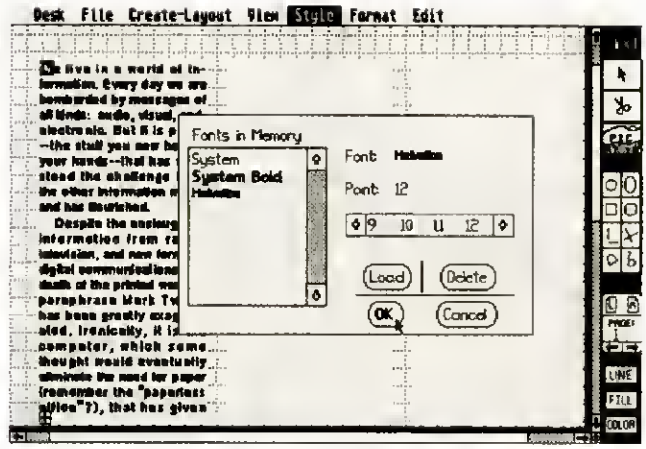


Figure 8a. Text in the 12 point Helvetica typeface as it appears on the screen . . .

Figure 7b. Samples of the type styles as they appear on paper when printed with a dot matrix printer.

Forms Creator
You can create a variety of forms quickly and easily with your Publishing Partner program. For example, you can create your own:

Letterhead
Invoices
Labels
Bumper Stickers
General ledger
Routing slips
Phone slips
Shipping logs
and much more!

nated. In fact, *Publishing Partner* goes way beyond most desktop publishing programs, offering a bewildering array of type enhancements, including backslanted, underlined, double underlined, light, mirrored (horizontally flipped), outlined, reversed (white-on-black), shadowed, tall (condensed), wide (extended), and upside down (see Figure 5).

In typesetting, *kerning* refers to the ability to set spaces between individual letters. This is desirable when two letters look better spaced closer together. For instance, in the word To, the letter o looks more comfortable tucked slightly underneath the overhang of the capital T (see Figure 6). This may not be obvious in text-sized type, but is readily apparent at headline sizes. *Publishing Partner* permits manual kerning of characters in an on-screen dialog box.

The Printed Word

All of this, of course, adds up to some very attractive page makeup features. But *Publishing Partner* is designed to create pages on paper, not on a video screen.

Currently, the program can handle most popular dot matrix printers, as well as laser printers. Newer 24-pin dot matrix printers, like the NEC P6 and P7, Toshiba 321, and Epson LQ series printers, will produce better quality output than, say, an Epson MX-80. And, true publication-quality output comes only from a laser printer. But there is a vast difference even in the quality of laser printer output.

While *Publishing Partner* can print to the Hewlett-Packard LaserJet Plus, the resulting resolution is only 150 dots per inch. (The program will also be able

Figure 8b. . . . and on paper actual size.

We live in a world of information. Every day we are bombarded by messages of all kinds: audio, visual, and electronic. But it is paper --the stuff you now hold in your hands--that has withstood the challenge from the other information media and has flourished.

Despite the onslaught of information from radio, television, and new forms of digital communications, the death of the printed word, to paraphrase Mark Twain, has been greatly exaggerated. Ironically, it is the computer, which some thought would eventually eliminate the need for paper (remember the "paperless office"?), that has given new life to the traditional craft of publishing.

to use Atari's own laser printer when it becomes available.) Fortunately, however, this is the first Atari program to support PostScript, which is quickly becoming a standard in electronic printing and publishing.

PostScript, written at Adobe systems by its founders, John Warnock and Charles Geschke, is a *page description language*—a true computer language, which is built into a number of laser printers, including the Apple LaserWriter, QMS PS 800, NEC SilentWriter, Texas-Instruments OmniLaser series, and the Qume ScripTen, as well as the Allied Linotronic high-quality imagesetters. It offers the first way of mixing graphics and text on a printed page with maximum flexibility and true typeset quality.

PostScript programs are written "on the fly" by other computer programs. When *Publishing Partner* describes a page to a PostScript printer, it actually writes from a dozen to a few hundred lines of PostScript code which, in turn, are interpreted by the computer inside the laser printer. The page that is produced by the PostScript printer only uses the on-screen representation of the page as a guide to create its own version (at 300 dots per inch) from its own library of special typefaces.

Even the least expensive PostScript laser printer is a major investment at over $4000. So *Publishing Partner* provides an alternative for users who want this high-quality output without the big pricetag. Using its own special smoothing routines for the creation of typefaces at any size, the program will produce good-quality output on dot matrix printers. This is usually good enough for checking and proofing the page.

PostScript code, though, can be written to a floppy disk instead of sent directly to a laser printer. Since PostScript code requires no special characters other than straight ASCII text, this file can be sent, via modem and telephone lines, to a site with a PostScript printer and fed directly to it. (Of course, if that site has an Atari ST connected to the printer, the floppy can be physically sent to the system, as well.)

This also means that a system comprised of an ST and *Publishing Partner* provides a path for true, very high-quality typesetting. An Allied Linotronic PostScript imagesetter produces type, not at the 300-dot-per-inch laser printer resolution, but at 1270 or even a staggering 2540 dots per inch.

Like any computer program, *Publishing Partner* is an idea that is constantly changing and developing. The version we used for review—version 1.00—has already been updated to fix minor bugs, provide several new printer drivers, and offer additional typefaces. We were unhappy with the limitations of using only a few typefaces—two (Helvetica and Times) with PostScript laser printers and two (Helvetica and the on-screen "system" face) with dot matrix printers.

Publishing Partner
virtually assures that Atari users won't be left out of the desktop publishing movement.

Soft Logik Corporation, publisher of the program, says it is working on more typefaces for dot matrix printers (Times Roman is ready), and a new version supports all the typefaces, including Bookman, Avant-Garde, Palatino, Zapf Chancery, and New Century Schoolbook, in the Apple LaserWriter Plus and QMS PS 800 Plus printers.

Furthermore, Soft Logik has just completed a version of the program that will automatically download PostScript typefaces not resident in printers by spooling the necessary typeface code from the ST disk drive to the printer. It is in the process of licensing a large number of original PostScript typefaces from Century Software, a Berkeley, CA, type design company, specifically for this purpose.

Wish List

The program cries out for other features as well, especially an Undo feature. Often, when moving blocks of type, graphics and other page elements, Undo can be a real life saver, particularly if you move a carefully-placed piece of artwork by mistake. This would come in handy, too, for sizing and cropping pictures which, at present, can't be changed back to their original state once they are altered.

The program manual, as mentioned, has a few problems, not the least of which is organization. After a "quick tour" and a longer tutorial, the bulk of the real information is relegated to a reference section, much of which is not illustrated. For a visually-oriented program like this, that is leaving too much to the user's imagination.

A word about system requirements: *Publishing Partner* works on a 520ST, but a quick glance at the system status checker (labelled Don't Look under the Desk entry on the menu bar) showed that the sample, two-page tutorial document furnished occupied about 60K of the 140K bytes available to the program. So the 1040ST seems like a better environment for *Publishing Partner*.

Finally, even though both monochrome and color versions of *Publishing Partner* are furnished, the color version is difficult to use, due to resolution problems. Indeed, the program will handle on-screen color and even print in color on special printers. But desktop publishing is, at present, a black-and-white medium. There are no color laser printers, and reproducing the (limited quality) color output from *Publishing Partner* would cost a small fortune on a four-color printing press. The real reason for using the monochrome, rather than color, video display is the absolute necessity of higher resolution.

Nonetheless, this is a good start. *Publishing Partner* virtually assures that Atari users won't be left out of the desktop publishing movement and, even now, is a very valuable program. Hopefully, it will even spawn some competition.

Few people remember, but author Mark Twain was a pioneer of automated typesetting systems. Twain invested enormous sums in a jury-rigged typesetting device invented by one James Paige, only to be left over $300,000 in debt when the device failed. (Ironically, it was Ottmar Mergenthaler who eventually bought all rights to the Paige machine.)

Twain, an old newspaper hand who remembered the drudgery of setting type manually, had faith in the revolution that automated typesetting would create.

So excited was Twain by the prospect of the automatic typesetter that he proclaimed, "It does not get drunk, it does not join the Printer's Union, and a woman can operate it."

And, his opinion of women notwithstanding, Twain would probably be quite happy to see how publishing has progressed. ■

System HD+ by Astra

2 DRIVES IN ONE !

LET'S TALK QUALITY!

* Top of the line components !
* Superior design !
* Extreme care in assembly !
* Scrupulous testing !

LET'S TALK CONVENIENCE!

* 20.8 megabyte hard-drive (formatted) !
* Double sided 3.5" floppy drive !
* Internal power supply !
* All in one neat, strong, housing !

LET'S TALK FEATURES...

* 36" hard drive cable lets you place unit conveniently, not on top of your keyboard !
* A single AC powercord replaces power supply and two powercords !
* Lighted front switch lets you turn unit on without a lot of gymnastics !
* Lighted busy lights tell you when your hard drive and floppy drive are working !
* Our small footprint (5 1/4"W, 4 1/4"H, 12"L) saves space on your crowded desk !

LET'S TALK WARRANTY...

We have taken extreme steps to insure the quality of these units. Therefore we feel confident enough to offer a ONE YEAR LIMITED WARRANTY covering defects in both parts and labor. This is four times as long as usually given in the industry!

LET'S TALK FUN !

Until you have experienced the fun and convenience of using a hard disk, you have missed a lot of the fun you expected from your computer. Faster access to programs and data make for more fun and less waiting !

LET'S TALK FREEBIES !

We supply complete formatting and partitioning software. (So does everyone else.) We also supply BRS,a unique backup and restore program that will back up your valuable files to floppies. This is a $39.95 value. And when you return your registration card to us we will send you a free "Mousterpiece" mouse pad !

LET'S TALK TURKEY...IT'S AT YOUR DEALER NOW !

---

*ASTRA SYSTEMS, INC.

2500 S. Fairview, Unit L
Santa Ana, CA 92704
(714) 549-2141

Printing By Laser

A brief introduction to the technology of laser printing

By JOHN J. ANDERSON

Do you really need a laser printer? At $5000, you can probably live without it. Nowadays, 16- and 24-pin dot matrix printers can produce some excellent print—really close to letter quality—and they're fast.

But have you seen the output of a laser printer? If you have, you may well feel that it could be tough to live without one. Though high-density dot matrix printers do a very good job indeed, they will never match the resolution or sharpness of laser output. Laser produced pages simply look fantastic.

The reason for this is simple: laser printers can print up to 300 dots per inch now, and that density is edging up as we speak. Even the best of dot matrix printers can muster no more than about 100 dpi. There is, therefore, no real way to compare the systems—except in relation to your own needs and desires.

Consider the fact that laser printers are quiet. The fact that they are fast, at least when printing multiples of a single page. The fact that they are incredibly versatile.

And that output looks so good.

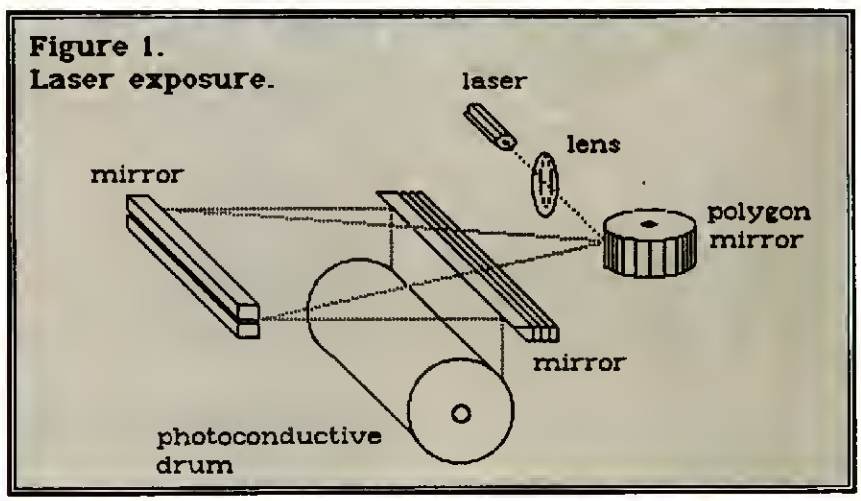
The design team at Atari has had its eye on laser printers for some time now. They have analyzed the factors that make the technology so expensive and, once again doing what they do best, found a way to crash the price barrier.

The sonic boom is going to shake the foundations of desktop publishing.

Atari has announced that it intends to make an all-out effort to enter the desktop publishing market with an advanced system that will cost only half what comparable products now on the market cost. To achieve this goal, the new Mega ST computer will be teamed with the new Atari Laser Printer. Atari's intention is to sell the complete system, including computer, printer, disk drive, and monitor, for under $3000. The 300 dpi laser printer itself will sell for under $1500.

The speed and RAM power of the Mega ST allow the laser printer to be software-driven, which means that Atari can reduce the cost of the printer and still achieve greater versatility. Under software control via the ST, the laser printer could conceivably use any page description language and any type font.

Figure 1.
Laser exposure.



How Does a Laser Printer Work?

Laser printers fall into the category of electrophotographic or page printers. These printers employ the fully developed technology currently being used within the printing units of plain paper copiers. The crucial difference between a page printer and an electrostatic copy machine resides in the image editing and the exposure unit.

With classic copiers, optical information in the form of an "original" is transferred to the photo-conductor drum by analog transmission in the form of a latent electrostatic charge image.

Printers and "intelligent" copiers, on the other hand, operate by way of a digital projection of electronic information to the photoconductive drum by a matrix of light dots—using one of various exposure techniques.

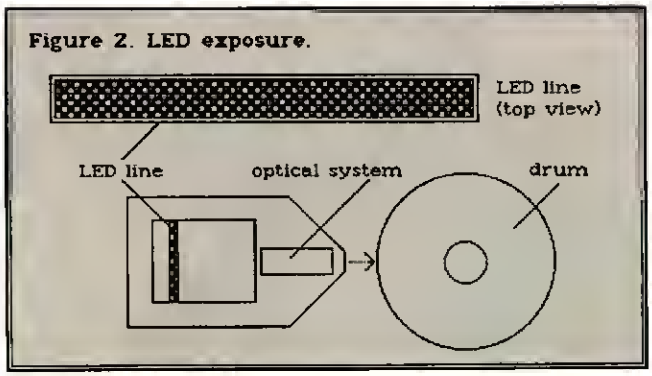
Currently, resolution of commercial page printers is 300 dots per inch, and resolutions of up to 800 dpi have already been achieved in the laboratory.

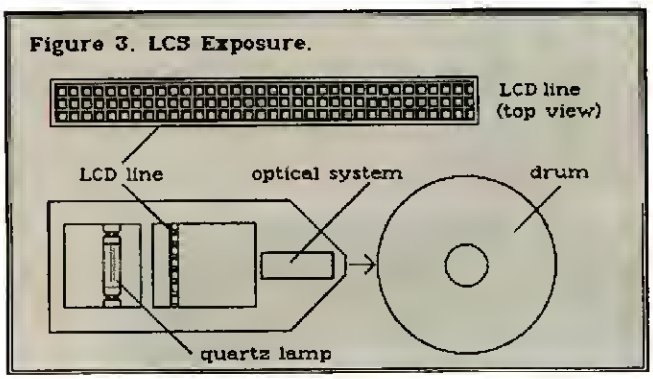
The printing unit is common to all page printers, and in fact many share an "engine" provided by a single maker, Canon. Other engines, all from Japanese sources, are now appearing. Some incorporate more discrete components, so replacement of consumable elements is made less costly. Others combine engine elements into single easily-replaceable cartridges.

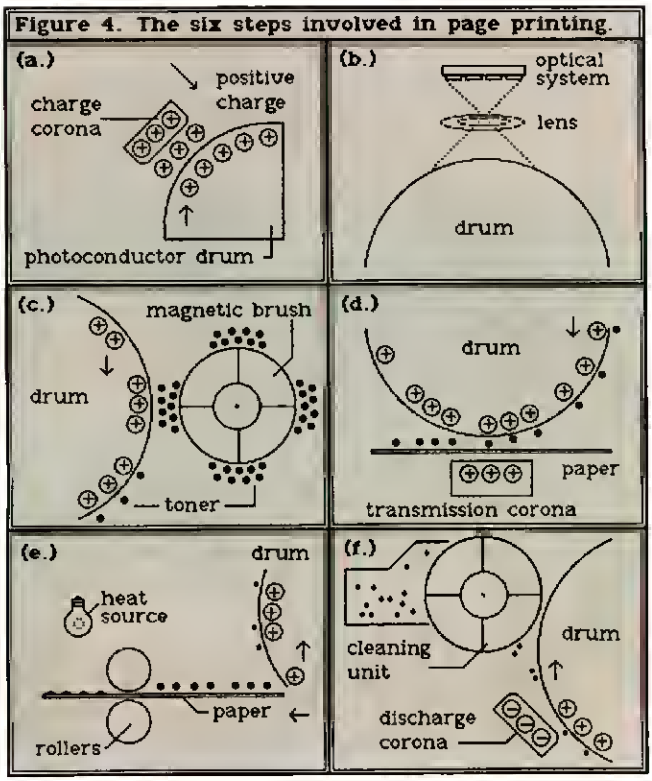
The Components

The components of a copier/printer engine include the following:

Photoconductor drum. The central element of the printing unit, the drum is







usually coated with selenium and acts as an insulator in the absence of light.

Corona. This component consists of a thin tungsten thread, which is fitted in a retractable and extractable plastic screen. When high voltage is applied by an electrical charge, it produces different charging and discharging conditions.

The printing unit is equipped with three different coronas. The *charge corona* charges the photoconductor in darkness with a positive electrostatic charge spread evenly over the entire surface of the drum. The *transmission corona*, which dispenses a higher voltage than the photoconducter, can then transfer the toner image from the photoconductor to the paper. The *discharge corona* discharges the electrostatic residual charge to the drum after the printing run or copy process is concluded.

Magnetic Brush. The magnetic brush is an electrostatically charged metal cylinder, which attracts the individual toner particles. These particles stand on top of each other and look rather like the bristles of a brush. Different charge conditions of magnetic brush and photoconductor drum cause the toner particles to jump over to the photoconductor.

All About Exposure

While the printing engines of different machines remain very similar, the exposure units differ widely. Three are most prevalent:

Laser is the most widely used technology today. An intensely modulated laser beam is deflected by a rotating polygonal mirror and directed over the photoconductor drum line by line (Figure 1). Precise and cost-intensive construction of the optical system and mechanical components is required to achieve a straight print line and constant dot size. This makes laser exposure units and their maintenance relatively expensive. Efforts are underway to reduce costs by using more moderately priced semiconductor lasers and simpler deflections.

New technologies, instead of deflecting a line at a time, transmit light information to the drum by more efficient means.

LED (Light-Emitting Diode) lines are one example; a large number of light-emitting diodes, each one of which can be activated separately, are arranged in matrix form (Figure 2). The activated diodes emit light, and the bright/dark pattern is then transmitted to the photoconductor drum by an optical system.

LCD (Liquid Crystal Diode) light switching lines make use of liquid crystal cells arranged in matrix form and situated underneath a quartz lamp (Figure 3). Each individual cell can be open or closed. The light passes through the open cells and is directed to the photoconductor drum as a bright/dark pattern by an optical system.

Do not confuse liquid crystal switching (LCS) with the liquid crystal display on your watch or pocket calculator. An LCS exposure unit sports liquid crystal arrays that can react about 200 times faster than the traditional LCD.

Putting It All Together

Figure 4 depicts the entire process of printing a page. The surface of the photoconductor drum is made light sensitive by the charge corona (a). The drum is then exposed with an image by any of the techniques described above.

A "latent" electrostatic image is produced (b). Next the toner particles jump from the magnetic brush over to the positively charged parts of the drum (c), and the laterally reversed image on the photoconductor becomes visible. As the paper gets a stronger charge than the photoconductor, it forces the toner particles to jump over to the paper where they are temporarily retained electrostatically (d).

Finally, under the effect of heat and/or pressure, the toner is fused to the paper (e). The drum is then cleared of any residual toner particles and electrostatically discharged (f). ■

The Future Of CD ROM

Industry experts convene to discuss standards and new developments in CD ROM technology

By JOHN J. ANDERSON

They descended upon waterlogged Seattle from all parts of the world. They came from every echelon of business, government, and computer science. They represented a hundred disparate opinions on the future of the technology. They represented a thousand vested interests. They represented millions upon millions of venture dollars, poised for bet on a shiny rainbow-silver roulette wheel as yet unproven in its promise.

This was a meeting of the pivotal minds at the Second International Conference on CD ROM, sponsored by Bellevue, WA-based Microsoft Corpo-

Microsoft Bookshelf—10 reference works on one CD.

Hitachi CD ROM drive.

ration. It was a key event for technical experts, application developers, consultants, corporate CD ROM users (and those who aspire to be), print publishers, online publishers, key equipment suppliers, and everyone else desiring involvement in the young technology of CD ROM.

It was an occasion for the most knowledgeable people in the fledgling industry to come together to discuss current developments and challenge the preconceptions of their peers. It was a technology pep talk, geared to stoke the burners that will eventually fuel the transformation of CD ROM—it is hoped—from promising development to multibillion-dollar industry. It also served as a forum for industry announcements and the unveiling of innovative research.

Microsoft Press has published a book called *CD ROM: The New Papyrus*. The idea is, one would imagine, somehow to underscore dramatically the significance of the advent of CD ROM. The analogy is perhaps even more apt than its author had intended. For papyrus was of great, almost holy value to the scribes and priests of ancient Egypt. It was not to be squandered on the needs of the masses.

So it is right now with CD ROM. The technology is prohibitively expensive, and its uses have thus been limited. The real task is to bring its innovative value down to earth.

Compact disk audio has taken the electronics industry by storm. Its acceptance and growth have wowed the experts. As a breakthrough technology, it offers some real benefits: superlative sound and no media wear. When player prices began to drop two years ago, the audio CD went through the roof.

Audio on compact disk is treated like any other chunk of digital information: it is reduced to a flow of on and off pulses. And audio is where the current consumer CD leaves off. But the fact is that any and all information can be represented in that same digital manner, and that is where CD ROM begins.

The Significance of CD ROM

So the potential of CD ROM in the future of computing is mammoth. What remains in question is how, and when, and what it will cost.

For while the advent of papyrus was undoubtedly significant, it pales in comparison to the advent of printing, called by Mark Twain "the incomparably greatest event in the history of the world." It was the advent of printing, not paper, that made books available to everyone.

Twenty-five years after the first book was printed in 1457, the printing capital of the world was Venice. And the busiest printer in the city was Aldus Manutius. Aldus knew that the real market for printed books was not in expensive, commissioned editions of the Bible or the Psalms, but in inexpensive little volumes that could easily be carried in a man's saddlebag wherever he went. So Aldus made his books small and cheap; his Aldine editions were the world's first pocketbooks, and they sold faster than he could produce them.

The CD ROM industry surely hopes that history will someday repeat itself. Parenthetically, Microsoft subtitled its 1987 Conference "Making It Happen."

If this is ever to be the case, some formidable obstacles must first be surmounted. And before we can talk about the cost of disk players, the cost of mastering, or the kind of content that will break through, we must first talk about standards.

■ **Toshiba XM-2000 CD ROM optical drive.**

The bitter battle of standards continues unabated and will continue for some time.

Standards

Many at the conference would have you believe that this question has largely been settled. Many at the conference stressed the need for cooperation between manufacturers in setting standards. Many at the conference embraced, in seeming altruism, standards put forth by competitors.

But it is an inescapable fact, repeated again and again over the course of the week, that the bitter battle of the standards continues unabated and will continue for some time.

In the beginning there was the Red Book, the standard for encoding audio tracks on a compact disk, which was developed jointly by Sony and Philips. The authority of these sources is hard to question—they brought us the compact disk in the first place. Acceptance of the Red Book is universal—no other CD audio standard exists. That is why any audio CD disk you buy will play on any audio CD player you buy.

The same companies then followed up with their own standard for CD ROM text, known as the Yellow Book. The latest entry from the Sony/Philips duo is the Green Book, which proposes a common standard for the interleaving of text, video, and audio. This standard is also called CD-I (for interactive).

While acquiescence has been heard from some quarters, and increasing pressure is exerted to comply, others have been observed dumping the Green

Book into Boston Harbor. Simply put, they find the CD-I standard sorely lacking.

Complicating the existing specifications for CD ROM is another critical consideration, called the *logical level*. Logical level considerations were addressed a little over a year ago by a coven of manufacturers at a casino hotel called the High Sierra on Lake Tahoe. They emerged with what is now called the High Sierra format. It specifies a standard for how data is represented on the disk—conventions, such as volumes, files, and records, and how they are used; how directories and paths should be organized; and how the volume and file structures should relate to the physical blocks.

Among those seated around the table at the High Sierra, placing a bet on a standard for CD ROM, were Apple, DEC, Hitachi, Microware, Microsoft, Philips, TMS, 3M, Videotools (now Meridian), Xebec, and Yelick.

The High Sierra standard proposes a common and known volume, directory, and file structure for all common operating systems. The idea is to avoid the incompatibility problems that splintered the magnetic software market from the start. The task of writing device drivers is left to suppliers of the operating system, rather than applications software developers or drive manufacturers. Publishers will create on CD ROM a single version of their data that will work with any computer, drive, or operating system, as long as that operating system has the necessary device drivers.

As an added result, device drivers do not have to be bundled on disk, but rather can reside on a floppy disk or in firmware on the interface board. Developers are, thus, never dragged into a battle over new formats.

The High Sierra group also specified "nested" logical levels, wherein simpler systems need support only a subset of the proposed standard; excess baggage need never be supported for its own sake.

Within the volume structure are a *system area* and a *data area*. The area reserved for system use is not specified by the High Sierra Group. The data area can be partitioned and features a series of volume descriptors that specify such things as volume identification, attributes, names and locations, and information needed to boot the system or application.

The proposed directory structure is a hierarchical tree providing as many as 65,535 directories. A path table affords access through the directory tree, and a maximum depth of eight levels in the directory hierarchy is specified. There is no theoretical size limit within the file structure, other than those mandated by restrictions on the volume set.

The High Sierra group described three kinds of files in an attempt to satisfy present and future application ranges: a *byte stream*, a *record structure*, and an *application structure*. The byte stream is the fundamental structure. It requires that recorded bytes in a file be treated as a single stream of contiguous bytes.

The record structure provides for a single predetermined partitioning of bytes in a single stream. The application structure is the integration of the first two; the goal is that it should provide for all application structures by specifying the byte stream and record structure.

The search for standards always faces a double threat. There is the danger of dawdling too long in setting a standard, causing chaos and weakening the industry in question. An example of this can clearly be seen in the magnetic software industry.

This danger is in turn tempered by the danger of committing too early to a standard that is weak. Witness the U.S. color television standard. It was set early on—much earlier than that of Western Europe. For that reason the systems used by the British and French offer much higher resolution and better color rendition.

There is the danger
of dawdling too long in setting a standard, causing chaos and weakening the industry.

Other News

Perhaps the most significant new announcement of the conference came from newly allied GE/RCA Laboratories, to the distress of both Sony and Philips. GE/RCA introduced an advanced, integrated video and graphics technology that provides digital full-motion, full-screen video, three-dimensional motion graphics and high quality audio capabilities from a single compact disk.

The new standard, known as Digital Video Interactive (DVI), integrates full-motion and full-screen video images, in contrast to any other format demonstrated to date. DVI technology can provide one hour of full-motion video along with high quality audio and indexing on a single disk.

At the heart of the DVI system are two proprietary VLSI chips that control pixel-processing and display-processing capabilities. This two-chip set is called the Video Display Procesor (VDP1 and VDP2). The VDP1 is the system pixel processor. Designed to run at 12.5 million instructions per second (MIPS), the VDP1 is a general-purpose programmable device that contains microcoded programs in its on-chip RAM. The VDP1 also includes special video instructions that give the chip the capability to perform video-based operations.

The VDP2 is the output display processor, which gives the DVI technology its resolution modes and pixel formats. VDP2 resolution ranges from 256 pixels to 768 pixels horizontally and up to 512 pixels vertically. VDP2 color depth pixel formats range from 8 to 24 bits per pixel, with a maximum of up to 16 million colors.

The VDP includes a computer bus interface that allows it to be used with a variety of system architectures. "We have announced DVI today in order to

solicit early feedback in the technology and its future commercial developments," said Arthur Kaiman, RCA Laboratories director of digital products research. "General Electric and RCA intend to explore a wide range of product and market applications for DVI technology," he continued.

When asked by *Atari Explorer* about DVI, Alan Kay said, "I don't yet know how good DVI really is, but I do know that CD-I is terrible." The search for standards continues.

Also at the conference, Microsoft unveiled its first general interest product in the field, a $295 compact disk called *Microsoft Bookshelf*, that contains the information found in ten reference books. The disk includes *Bartlett's Familiar Quotations, The 1987 World Almanac and Book of Facts, The Chicago Manual of Style, The American Heritage Dictionary, Roget's II Electronic Thesaurus, The Houghton-Mifflin Spelling Verifier and Corrector, Forms and Letters, The U.S. Zip Code Directory, Houghton-Mifflin Usage Alert,* and *Business Information Sources.* *Bookshelf* is capable of operating with 14 popular word processing programs for the IBM PC and compatibles.

A new CD ROM drive *features an audio mode to allow conventional playback of current format CDs through home stereo equipment.*

In other developments, a new CD ROM drive shown by Amdek features an audio mode to allow conventional playback of current format CDs through home stereo equipment. This is the first instance of such linkage, which is viewed as a crucial sales feature for consumer level players. Amdek marketing manager Jerry Benson announced that Sears Business Systems Centers will carry the CD ROM drives.

If there was one fact to emerge from the Second International Conference on CD ROM, it was that the jury is still out—perhaps we'll know more by the Third International Conference on CD ROM. Or the fourth. Certainly the technology itself is most impressive.

But in closing his address, Kay reminded the assembly of conference participants that "none of these technologies is central—what is central is the notion of the computer as something that is much more than any medium has ever been before. The Greeks held that the visual arts were the imititation of life. Well, the computer arts are the imitation of creation itself. Cesare Pavese, the Italian writer, said, 'to know the world, one must construct it.' But to construct it dynamically will require all the technology, ingenuity, and aesthetics we can muster." ■

ComputerEyes

At last, a color video digitizer for the ST

Among Atari ST owners with a penchant for graphics, no single product has been so long-awaited as a quality video digitizer. Video digitizers for the ST have in fact hit the market in the past, but until now have proved unworthy of the hardware they purported to support.

But things are different now that ComputerEyes from Digital Vision has appeared on the scene. ComputerEyes is a full-color video digitizer for the Atari ST, which allows you to easily capture images in full color from any standard video source—camera, VCR, or laserdisc. Supporting all of the graphics capabilities of the ST, ComputerEyes represents a unique and relatively low-cost means of capturing monochrome or color graphics on an ST display.

The package consists of a hardware unit that plugs into the cartridge slot of an Atari 520 or 1040ST and a software program that drives the unit. Under simple, mouse-driven software control, the video signal is scanned and images captured in the graphics memory of the computer. Capture time is just six seconds for the lo-res 16-color or -intensity level mode, 12 seconds in the mid-res four-color mode, and 24 seconds in the 640 × 400 hi-res monochrome mode.

Applications for ComputerEyes include acquiring images for the popular graphics drawing programs like *Degas* and *NeoChrome*, capturing real-world images for desktop publishing and other printing programs, image databases, and even industrial applications like pattern recognition, spatial measurement, and security. ComputerEyes images can also be transferred to T-shirts for fun or profit.

A unique feature of ComputerEyes is that, once acquired, images can be *adjusted*, or fine-tuned, using the system software to obtain the desired image or effect. Images can be displayed and redisplayed in levels of gray (two, four, eight, or sixteen intensities) or full color. Changes in the brightness, contrast, and red, green, and blue parameters can be made after image acquisition to yield a sensational display. You can also specify color coarseness and the number of colors or intensities in the image.

Acquired images can be saved to disk as standard *Degas*, *Degas* Compressed, or *NeoChrome* compatible files. The software package features the ability to translate screens to and from either format.

How It Works

ComputerEyes is a carefully designed combination of hardware and software, neither of which has any worth without the other. The system performs a slow scan on a video signal present at the video input connector on the hardware module. During every vertical scan period, the system takes in 200 samples, or one for each of the Atari ST graphics rows. Thus, one column of pixels is stored for every vertical scan, or 16.6 milliseconds.

The point during each horizontal scan period at which the samples are taken is delayed from the horizontal sync by an amount that is very slowly increased as the scan progresses. The result is that successive columns of pixels are acquired, beginning with a short delay from the horizontal sync (left side of image) to a long delay (right side).

A video sample consists of three analog voltages that represent the intensity values of the red, green, and blue components at a particular point in the video image. These voltages are multiplexed into a 6-bit analog-to-digital converter, the output of which is read by the Atari ST. The software stores these raw image data values away in memory during the scan. It then maps the values to the Atari ST graphics memory based on the type of image that is being acquired.

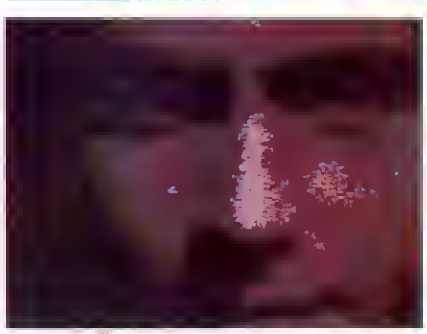
The ComputerEyes hardware consists of a single printed circuit board mounted in a plastic enclosure that plugs into the ST cartridge port. This board contains the circuitry required to perform the slow-scan image acquisition. The software running in the ST controls the acquisition of the image.

The ComputerEyes control panel consists of a collection of on-screen buttons and slider controls that allow you to control operation of the package with the mouse. In fact, the only time you ever have to use the keyboard is to enter the names of image files that you want to save. You can view the image currently in memory at any time simply by pressing the right mouse button. Pressing the same button again returns you to the control panel screen.

A computer's eye view of the shuttle.

System: Atari ST
Price: $249.95
Summary: An extremely capable color video digitizer
Manufacturer:
Digital Vision
66 Eastern Ave.
Dedham, MA 02026
(617) 329-5400

By JOHN J. ANDERSON



Collins as Tweaked by Anderson.

Peter Anderson—a Father's Day Tribute.

Making Adjustments

Clicking on the Capture button causes the video signal present at the input connector of the ComputerEyes module to be scanned and displayed on the ST monitor. The system uses the current control panel settings of resolution, scan speed, and gray levels to acquire and display the image. While the video input is being scanned, a message to that effect is displayed on the screen. When the scan is complete, a bell tone is heard (handy, if a camera is being used, to let a subject know when he may move). The screen is then painted from top to bottom with the image that has been captured and processed according to the settings of the control panel.

If you are acquiring a color image, the system takes another few seconds after the scan period to analyze the image to determine the color palette it will use to paint the image. It displays a message on the screen to this effect and then paints the screen.

Clicking on the Adjust button causes the last-acquired image to be repainted on the screen using the current settings of the control panel. After you capture an image, you can modify its appearance by adjusting the brightness and contrast sliders, the gray-level settings, the red, green, and blue sliders (for color images), and the number-of-colors option (also for color images). After you make a change to any of these settings, you simply click on Adjust to display the result. You can cancel the repainting of the screen before the process is complete by pressing the right mouse button.

If you are adjusting a color image, the system takes a few seconds after you click to analyze the image, in the process automatically determining the color palette it will use to paint it. It is important to understand that Adjust recalculates the palette and the value at every screen pixel location from a large table of raw image data created by performing a capture. This means that you can adjust the appearance of the image as many times as you want without losing the original image data. The image is not rescanned unless you click again on the Capture button. This is a very powerful approach and allows you to optimize the characteristics of every image that you capture.

If you load an image from disk, it cannot be adjusted as a freshly captured image can. This is because it is a picture file, and does not contain the raw image data created by performing a capture. The Adjust button and slider controls are in fact disabled when you load an image from disk; they are enabled only after you capture an image. Unfortunately, raw data sets themselves cannot be saved to disk.

The Whtbal button is provided as a means of initially setting the red, green, and blue sliders to settings that will give generally a realistic color balance. It works like the white balance adjustment button on a color video camera.

To use it, you provide a video output as close to white as possible by using a white translucent lens cap or by aiming the camera at a white object, such as a sheet of paper. You then click on the Whtbal button. This causes the system to spend a few seconds analyzing the content of your video signal. When it is done, it automatically updates the settings of the red, green, and blue sliders to the appropriate values.

If you capture an image before doing a white balance, the system calculates RGB settings based upon the actual image. This may result in an image with unbalanced color content, but the image will definitely be displayable. You can always do a white balance even after the image is captured, or adjust the R, G, and B sliders manually, and click on Adjust to redisplay the image.

If you are capturing images directly from a videotape recorder, the Whtbal button won't really be of any benefit. You are better off just letting the system calculate RGB values and readjusting them manually as you see fit.

There is no Whtbal button on the monochrome control panel, because the balance of red, green, and blue is unimportant in monochrome images. Also, if you are capturing images in one of the gray scale modes (as opposed to color), the RGB settings are entirely disabled.

The resolution area is a pair of buttons that allows you to select the resolution (lo-res or mid-res) of the image that you are about to capture. It has no effect on images that you have already captured.

Normally, it takes six seconds to capture a lo-res image and twelve seconds for a mid-res. However, if you wish, you can choose a slower scan rate (12 seconds for lo-res and 24 seconds for mid-res) to produce images with somewhat better resolution. The reason for this is rather technical, and has to do with the *interlace characteristics* of composite video signals. Essentially, you can trade capture time for a certain improvement in image quality. This improvement can be most readily noticed as somewhat less jaggedness along lines that are close to horizontal.

Brightness, Contrast, and RGB

The brightness and contrast sliders work exactly like their counterpart controls on a television set. Increasing the brightness slider produces an image with higher intensities. Increasing the contrast yields an image with greater differences between light and dark areas.

A small button located near the brightness and contrast sliders is labeled Rst (for restore). If you click on this button, you will restore the brightness and contrast buttons to their original values as calculated by the system when the image was captured. This is handy for starting over after you have been fooling around with an image for a while.

The red, green, and blue sliders are provided to let you modify the color content of the image that you capture. This can be either to fine-tune the color balance for the most realistic results or to modify the balance to achieve special effects.

Unlike the brightness and contrast sliders, the R, G, and B sliders are not changed each time you perform a capture. This is because although it is correct to change the brightness and contrast settings to compensate for lighting and other conditions, you will not want to change the color balance once you have it adjusted correctly.

It will probably take a little practice before you get a feel for the effects of various R, G, and B slider changes. For example, it is not necessarily obvious that increasing both red and green will make an image appear more yellow. As

the raw capture data remains in RAM memory, it does no harm to experiment with various changes to see what the effects are.

Grey Level Options

The gray levels area gives you dramatic control over the color palette used to display images that you capture. Note that by capturing an image, which results in a large table of raw image data, you can change the gray levels (or color) as often as you wish and redisplay the same image in several different modes. When you are happy with the way an image appears, you can then save it to disk with the palette and screen pixel values as they last appeared.

Descriptions of the various gray level options follow:

•2: Only two palette colors are assigned—black and white. Essentially, every point in the image is determined to be either above or below a certain threshold, and the image is displayed entirely as areas of black and white. If you increase the brightness slider setting, more of the image will appear white, and vice versa. The contrast and red, green, and blue sliders are disabled, because they have no effect on the appearance of the image. This type of image is ideal for line drawings, logos, and other applications where the image being digitized already has a lot of contrast.

•4: Four palette colors are assigned—black, dark gray, light gray, and white. The brightness and contrast sliders can both be used to modify the appearance of the image. The red, green, and blue sliders are disabled.

•8: Eight palette colors are assigned—all eight of the pure gray intensity levels of which the ST is capable, ranging from black to white. The brightness and contrast are enabled; RGB sliders disabled. This entire selection is disabled in mid-res, because only four palette values are supported by the ST in this mode.

•16: Sixteen palette colors are assigned. These include all eight of the pure gray intensity levels of which the ST is capable, plus one additional color between each pair of gray levels, which is obtained by increasing the R, G, or B value by one. The result is a 16-level monochrome image with a distinctive tint. Red, green, or blue can be used to tint the image. Brightness and contrast are enabled; RGB sliders disabled.

•Color: Up to 16 palette colors are assigned (four in mid-res). You can choose to use fewer colors, saving one or two for your own use, or for special effects. The ComputerEyes software uses a sophisticated "voting" technique that determines the palette based on the color content of the raw image data and the settings of the various slider controls. Brightness, contrast, and red, green, and blue sliders can all be used to modify the appearance of the image. Unlike color digitizers that require three passes of a monochrome scan under red, blue, and green filters, ComputerEyes does it all, in full color, in a single scan (if you have provided a color input).

Another color setting, Separation, is also extremely handy for dealing with color input. As an example, let's imagine scanning an American flag. This video input consists of exactly three colors, so we set the number of colors to three. Then, to clarify the scan, we will push the Separation setting. The higher the separation setting, the more ComputerEyes will ignore colors that are close to each other, skipping on to colors with more differences between them. Experimentation with the separation setting will yield superior images.

For the most part, the description above applies to the monochrome monitor version of ComputerEyes as well. The monochrome control panel contains a relevant subset of the color panel. Monochrome images have a resolution of 640 × 400 pixels, each of which can be either on or off (white or black).

Two Scanning Modes

ComputerEyes has two scanning modes:

•**Normal** is a very high contrast mode, similar to what is obtained by setting gray levels to 2 in the color control panel. Every point in the image is determined to be above or below a certain threshold, with the resulting image being composed of solid areas of black and white. Due to the high resolution of the ST when used with the monochrome monitor, this capture mode yields images of exceptional clarity and sharpness. It is excellent for capturing such images as line drawings, font sets, etc.

Because there are so many pixels to be scanned, the normal mode takes about 24 seconds to capture an image. The brightness control is enabled in this mode, but the contrast control is not.

•**Dither** mode allows you to represent gray tones on the monochrome monitor. Here, gray levels are represented as varying dot intensities. The term *dither* refers to the technique that is used to determine the patterns of dots used to form the gray levels. Because there is a slight smearing effect inherent in the dithering process anyway, a six-second scan is used without loss of image quality. Brightness and contrast sliders are active in this mode.

As a graphics tool, ComputerEyes is superlative. Our experimentation with the device resulted in excellent scans in all resolutions, especially 16-shade tinted mode. Our experience with color was less impressive, but that may have had more to do with the inadequacy of our lighting conditions than anything else.

It is unfortunate that the raw video data set cannot be saved to disk, as is possible with Thunderscan for the Macintosh. But to be fair, MacVision from Koala Technologies—the equivalent video digitizer for the Mac—is blown away by the capabilities of this package.

ComputerEyes packs a lot of power into a little package at a list price of $249.95. This may seem like a lot, but it is actually extremely reasonable for the features it provides. ComputerEyes software is not copy-protected, and it requires TOS in ROM. ■

ST Self Portrait.

Atari's Technical Support guru answers your questions about Atari computers

Question Mark

By MARK JANSEN

Q Will it hurt anything to remove a disk from the drive while the drive is running?

A Removing a disk while the "busy" light is on is not generally a good idea. If you do so, you risk interrupting the drive while it is writing onto the disk such vital information as a filename, the position of a file on the disk, etc. This can cause files to be damaged or lost or the entire contents of the disk to be corrupted. In general, then, you should make sure that the busy light is off before removing a disk.

It must be said, however, that "popping" a disk can sometimes be the only way out of a jam. If, for example, you have accidentally given the computer a command that will cause erasing or overwriting of a valuable file, a quick hand on the eject button may be your only hope. Note, however, that you are risking the contents of your entire disk (and, to some extent, the safety of your drive, as well) by trying to beat your computer to the punch.

Q I have an older 800 with less than 48K of memory. Can I still purchase additional memory modules for my machine?

A Yes, 16K memory modules are available from Atari Customer Relations for $19.95 each plus $2.50 for shipping. Be sure to specify that you want a 16K RAM module for the Atari 800 computer.

Q I have some older 8-bit software that works fine on an Atari 800, but refuses to load on my XL or XE computer. What should I do?

A The Basic programming language was provided in cartridge form for the Atari 400, 800, and 1200XL; all other Atari 8-bit computers have Basic built in. Some programs refuse to run with Basic present, a problem 400, 800, and 1200XL owners can solve by removing the Basic cartridge before switching on the machine. Owners of more recent Atari systems can temporarily "remove" Basic by holding down the Option key while booting.

XLs and XEs also come with a revised version of the Atari operating system. Certain older programs are incompatible with this OS and may freeze up, behave unpredictably, or simply refuse to boot, when you try to run them on newer 8-bit systems. By pre-booting with an Atari Translator Disk, you can make your XL or XE series computer "look" just like an Atari 400 or 800 to 95% of this older software.

To order the Translator Disk, send a check or money order for $9.95 plus $2.50 for shipping to Atari Customer Relations, Attn: XL Translator Disk, P.O. Box 61657, Sunnyvale, CA 94088.

Q I have DOS 3.0, which came with my 1050 disk drive. How can I get Atari's latest version of DOS, version 2.5?

A Simply send your original DOS 3.0 disk to Atari Customer Relations at the address above. If you mark the envelope "Attn: DOS 2.5," we'll send you a DOS 2.5 disk and a mini-manual sufficient to get you up and running. If you would prefer to receive the full-sized DOS 2.5/1050 Owner's Manual, which contains more detailed information and is a useful programmer's reference, mark the envelope "Attn: DOS 2.5/1050," and enclose a check for $10.00 plus $2.50 for shipping.

Q What are "desk accessories?"

A Desk accessories are small programs that reside in the memory of your ST and can be called up to perform various tasks even while major GEM applications are running. Examples of desk accessories are the Control Panel and VT52 terminal emulator, both of which come free with the ST. Many other kinds of desk accessories are available, ranging from desktop calculators for handling quickie math problems to digital clocks scheduling aids, phone dialers . . . you name it!

Desk accessory programs, the names of which all terminate with the extension .ACC, are automatically loaded from your boot disk (or the root directory of your hard disk drive) when you turn on or reset your computer. Up to six accessories may be loaded at once. Their names are placed in the Desk menu—the leftmost item of the desktop menu bar. To activate an accessory, you just double-click on its name.

Note that in constructing their own menu bars, certain GEM programs eliminate the Desk item in the desktop menu, changing it to some other symbol. *1st Word*, for example, changes the word Desk to an Atari logo. Accessories are usually still available, however, under the new menu heading.

Accessories are not available when running programs that do not support the GEM menu bar. This is one reason that most Atari owners insist on software that makes full use of GEM.

Q I am using the *AtariWriter* cartridge with a non-Atari printer. How can I take full advantage of the printer features?

A The easiest way to make the *AtariWriter* program work correctly with a non-Atari printer is to get a *printer driver* for the program.

These drivers are available from Atari expert Gary Furr. Send a cashier's check or money order for $10.00 and a description of your printer (manufacturer, model, etc.) to:

Gary Furr
P.O. Box 1073
Mountain View, CA 94042

Q Where can I get a list of all the software available for the ST?

A The International Software Catalog from Atari lists hundreds of programs from various companies, along with capsule descriptions of each. To receive the catalog, send a check or money order for $12.95 plus $2.50 for shipping to Atari Customer Relations, Attn: International ST Software Catalog.

If you have question about your Atari computer, please send it to Question Mark, Atari Explorer, 7 Hilltop Road, Mendham, NJ 07945. ■

Aegis Animator ST

Aegis Development releases a set of the most sophisticated and powerful animation tools ever created for a microcomputer.

By JOHN J. ANDERSON

Picture this: Two kites side by side on a lo-res screen, one red, one blue. They move independently as if caught by the wind. In an instant the one on the left begins to twirl head over heels, while the one on the right spins on its tail. Then they trade spins. They rotate—first together, then independently. The blue kite next slides smoothly right in front of the red kite, and as if in retaliation, the red kite wings swiftly around to the left to come in once again ahead of the blue.

Even though they overlap each other now, each kite begins its own entirely independent antics, tossing, turning, spinning. As they continue their dance, we zoom in closer and closer, until we seem to actually "break through" the red kite, and it falls to the earth.

Only the blue kite is left now, and it freezes solid, alone in the center of the screen. From it, after a short pause, jump four new kites, exact clones of the original blue kite, but in yellow, green, orange, and purple. The kites assemble into a single, perfectly stacked multi-kite, executing complex spins and turns.

Suddenly they all burst apart, spinning around the screen in unison, as we zoom closer and closer to them. They form a chorus line, kicking out the points of their tails to the "beat" as they exit, screen right.

Yes, folks, this is for real, in real time, with real quality. Total time spent to create the animation described above: five minutes. Package: *Aegis Animator ST*. Verdict: Wow!

If graphics are a part of what matters to you in your exploration of the powers of the Atari ST, you ought to have a good look at *Aegis Animator ST*. For although a good imagination may be able to give the description above some life, only a good processor can slap it on a screen with tangible speed, polish, and rock-solidity. Which is, after all, what Atari graphics are for.

Aegis Animator ST is an extremely powerful animation tool. It can produce objects in 16 simultaneous colors from the full range of 512 available in ST lo-res. It can animate those objects by changing their shape, size, or color, or by dragging, rotating, or plotting a path for them on the screen.

It allows you to create up to six separate "scripts," and cut or splice them together on a single storyboard. It allows you to import blocks and backgrounds from *NeoChrome* and other paint packages.

In short, it provides a set of the most sophisticated and powerful animation tools ever created for a microcomputer. And for all its capability, it is easy to use.

System: Atari ST
Price: $79.95
Summary: An exceptionally powerful animation tool
Manufacturer:
Aegis Development
2210 Wilshire Blvd.
Suite 27
Santa Monica, CA 90403
(213) 392-9972

Three Kinds of Animation

With *Animator ST*, you can produce three different styles of animation: *color cycle*, *cel*, and *metamorphic* (which also makes simple 3-D modeling possible). The metamorphic capability is what makes the package so special. It uses the concept of *tweening* to control movement by creating an image and then changing its position or shape on the screen in a segment of time. Each time segment is designated as a tween. You decide what the shape should look like at the beginning of a tween and what you want it to look like at the end of that tween. Software creates the intermediate steps, while you sit back and watch. The real power of *Animator ST* is its ability to handle the gradual (or not-so-gradual) change from one position to the next.

This concept of metamorphosis differs from traditional cel animation, which can also be done with *Animator ST*. To cel animate a bird flying, you would first draw a bird with wings spread, let's say, in the up position. In conventional film animation, the wings would be painted on one piece of clear celluloid and laid over the bird's body, which would be painted on another piece. For the next position, instead of redrawing the whole bird with a new wing position, you would use the body from the first frame but with the wings repainted a little further down on a new piece of celluloid laid over the bird's body.

This process is repeated until the wings are all the way in the down position. The same images can then be used in reverse, moving the wings back into an up position. At the same time, you would want to move the bird (both pieces of celluloid) forward in relation to the background to give the effect of forward motion. When these images are played back in rapid succession, the bird seems to flap its wings and fly forward.

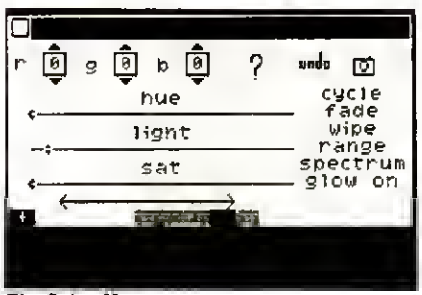
In the metamorphic mode of *Aegis Animator ST*, the bird could be made to fly by changing the shape of the wing, say from its up position to a position in the middle. This would occur in the first tween. Then, the shape of the wing would be changed from its middle position to its down position in the next tween. At the same time, the bird could be pulled forward a little in each tween to create the illusion of forward motion.

When the two tweens are replayed, the wing will smoothly and automatically shift from its up position to its down position in one movement, and the bird will move ahead.

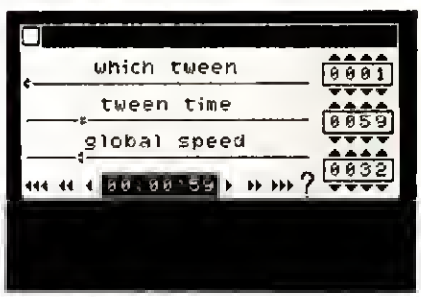
Although you can import images from popular paint packages, you cannot use metamorphic shape-changing techniques on those images. The alternative is to have *Aegis Animator ST* create simple cel animation using previously created paint images. You would, for example, load in the paint images for the bird's body and all wing positions.

Another useful feature of *Animator ST* is color cycling, which causes the color of objects on the screen to change through a range of designated colors. Color cycling is very handy when you want like to inject an imported background with a bit of motion. Using color cycling, you can make a static backdrop appear animated. For example, you can

The Main Screen.



The Color Menu.



The Time Menu.

Changing the size of a polygon.

Rotating a polygon around a Y axis.

color cycle a picture of a waterfall to simulate falling water. This feature is duplicated by both *NeoChrome* and *Degas Elite*.

But it is metamorphic animation that makes 3-D modeling possible. For example, let's say you wanted to simulate the rotation of a cube. To make a side of the cube appear as if it were coming around to the front, you would use the metamorphic tools to change the shape of that side, making it grow and change perspective. Simultaneously, you would make the side in front appear smaller and more distant in perspective to simulate its travel to the rear.

Another method involves creating each side of a cube by specifying points that are on planes other than the screen plane and then rotating the sides through the planes.

With all of these techniques combined, *Aegis Animator ST* becomes the most powerful animation tool yet released for the Atari ST computer.

Using ST Animator

When you boot up the animator program, a conventional GEM menu bar appears across the top of the screen. Its headers are Menu, Make, Move, Pick, Time, and Color. Under the Menu header, the selections include: Undo, which allows you to undo your previous action; Help Bar, which inserts a prompt line at the bottom of the screen to aid beginners; Fast Menu, which pops open an icon-based window, allowing you to perform your most often-used tasks quickly and easily; Storage, which allows you to read and write animations, graphics blocks, backdrops, masks (monochrome backdrops), and strips to and from disk; Storyboard, which allows you to splice sequences together; Color, which enables various colors, palettes, and their associated effects; and Time, which allows you to tweak the speed of various tweens as well as entire sequences.

Under the Make header, you can create shapes to be used in metamorphic animations. Beginning with a line, polygon, circle, star, or block, you can create nearly any shape imaginable. A Clone selection allows you to duplicate existing shapes. Destroy allows you to remove shapes.

Another pair of commands, Insert and Exchange, gives you the ability to replace existing shapes or to add new facets to them.

The next header choice is Move. This group controls the different methods of moving an object about the screen. You can move objects sideways, in front of, or behind other objects; rotate across the screen plane; rotate across a vertical or horizontal axis; make bigger or smaller; move along a defined path; change color, choose whether a shape is filled or outlined; and make metamorphic changes (change the shape). Remember, the program itself will see to the smooth transitions between your tween start and end points.

To the right of Move is Pick. This menu choice controls what items or portions of items on the screen you will alter; it determines specifically which items on the screen will be affected by another command. You can select all items, a number of them, one of them, a portion of one of them, or individual points.

The next selection is Time. The choices under Time perform functions like moving to the next tween, replaying the animation created in the current tween or in all tweens, and playing the animation in a continuous loop. The last three items are preferences. The first, Ghost Mode, displays the entire animation in outline form only. This is useful for seeing objects that are currently behind other objects on the screen. The other choices, At End and At Start, allow you to choose whether you view a tween as it exists at its beginning or as it is when the animation ends.

The last choice on the menu bar is Color. It allows you to choose the colors you will be working in when you create the different objects on the screen.

An Animated Example

So. Let's create a quick animation. First, create a shape: choose Polygon from the Make menu in the color of your choice. Click the left mouse button to drop each angle on the screen, then the right mouse button when you are done.

Now let's animate the shape. Choose Sideways from the Move menu; then drag the shape to a new position. You can review the tween you have just created by selecting Play Tween from the Time menu. To differentiate this action from the next, select Next Tween from the Time Menu.

Next we'll make the shape spin by selecting Y Rotate from the move menu. Set the cursor down where you want the rotation axis to reside. Then move the shape through the rotation you desire. If you want the shape to move on the screen as it spins, simply choose Sideways from the Move menu and select a new position for the shape before you click on Next Tween.

The actions will now take place simultaneously. For a finale, choose Next Tween and rotate the object in the

screen plane while making it recede to a small dot. Choose Big/Small from the Move menu, and make the change alongside the rotation command.

That's it! It has taken less than a minute to create a sophisticated animation that would have required an hour or more of work using conventional animation techniques. And yet, *Aegis Animator ST* has the kind of depth that will keep graphics aficionados in thrall for quite some time.

The process of discovery with the program is a source of delight, and the result of your experience will be ever more sophisticated animations.

Once you have broken into the mechanisms of *Animator ST*, you will find that a bit of patience and attention to detail really pays off. Like any art form, animations created with this program will improve as you invest additional time and effort in them.

Plan Ahead

Some animations can just be created on the fly without any advance work, but more complicated ones will need a lot of preparation. Backdrops, text scrolling on the screen, and paint images that move about all require advance work in a paint program.

Another type of animation that requires some planning is that in which an object suddenly appears from behind another. Polygons exist on successive planes behind one another as they are made. So if you want your space creature to grow antennae smoothly, you must include them when the creature is first created and maintain them as an inactive polygon behind the rest of the creature until needed. This requires a little forethought, because going back later and inserting the antennae will cause them to appear on top of what is already there.

One way to get around some of these problems is with the Save and Storyboard features. You can create an object independent of the end animation and save it to disk as a polygon, cel, backdrop, or mask. Then you can go back and build the final animation piece by piece. When each element is needed in the animation, load it from the storage window.

The Storyboard provides a similar capability without the flexibility of bringing in a single object at the right moment. Starting tweens can be perfected in one window, the preliminary adjusting movements cut out as a tween, and the final product moved into a new window ready to begin. Or, using the space creature antennae example, a copy of the current animation can be spliced into a clean window and the antennae added in that version of the script.

Planning is also useful in achieving smoother animations. Speed is the main consideration in avoiding jerky movements. It takes *Animator ST* longer to draw a polygon than a mask or a cel, and color changes require even less processing time. For these reasons you want to animate as much as possible with masks and cels, saving the metamorphic shapes for key roles and enhancements to the raster images.

For example, a tree can be made by creating the trunk and some leaves in *NeoChrome* and the rest of the leaves or branches that need to move in *Animator ST*. The *Animator* leaves are drawn right on top of the raster image, and even though just a few of the leaves move, it appears as if the whole tree is animated. Add to that a subtle color shifting of raster leaves, and the effect is complete with only a few polygons.

Another way to get good control of the speed and smoothness of the animation is to use plenty of tweens. The speed is adjustable in each one, so you have more control over individual movements.

Once an animation is complete or nearly complete you will want to go back and watch the whole thing several times, looking for places in need of minor adjustments. The process of making changes in an existing sequence is known as "upstream" editing.

It is important to consider the future of an object when doing upstream editing, taking care to make the proper adjustments to compensate for any changes. This is the only way to be sure that the transitions in your tweens will remain smooth.

***The process of discovery** with the program is a source of delight, and result of your experience will be increasingly sophisticated animations.*

Negative Nits

The package does have some eccentricities and shortcomings, but they are minor when compared with the unbridled power of the software. The major problem with *Animator ST* is the implementation of the GEM interface, which can be, well, somewhat erratic at times.

It is possible to pull down a menu, for example, highlight a selection, then click right through it, placing a dot on the screen "underneath" that menu, rather than choosing the highlighted selection. Mouse button clicks can seem "sticky," and it is sometimes difficult to get a click to register. When your mind is deep in the considerations of an animated sequence, problems like these don't seem so trivial.

For this reason it is in your best interest to learn the Fast Menu as fast as you possibly can. It is less bothersome to control and really is faster to use.

In my experience with the program, it was almost always possible to "undo" errors of the GEMish kind, though once I actually did lock up in the middle of animation. Needless to say, in rebooting I lost the sequence in progress.

Animator ST also lacks the transition effects of a slide show program, and it is not possible to fade or dissolve between raster images. If it is a slide show program you seek, *Aegis Animator* is not the product you want.

Then there is the fact that *Degas* images must be converted to NEO format before they can be accessed by *Animator ST*. This is not an insurmountable problem, but it would really be nice if *Degas* files could be read directly into the program.

Another criticism of the package, though it may not be fully justified, is that the program is limited to only the lo-res mode of the ST. It is undeniable that many sophisticated effects are available in this mode, and its resolution is more than acceptable for most applications. On the other hand, if some subset of *Animator ST* were available for higher res modes, the utility of the program would be enhanced.

Despite these reservations, most serious of which is the occasional lack of cooperation from the user interface, *Aegis Animator ST* is one of the most impressive graphics packages available for the ST series. If you are into ST graphics, it is a must for your software shelf.

The package is unprotected, so backup copies can be made, and a player program is included with the package, so animation programs can be easily distributed. Aegis Development invites owners of *Animator ST* to distribute the player freely with their own animations. ■

Puzzles & Problems

By DAVID H. AHL

Most of our problems this issue require you to write a computer program. There is even a small contest you could win if you design the best program to solve one of these problems.

Answers are on page 50.

The Keyring Problem

You have a circular keyring with five keys on it. Your challenge is to engrave a number on each of the five keys so that by summing the numbers on adjacent keys, you can achieve any total from 1 to 21. For example, if the keyring contains keys with the sequence 1-2-4-X-Y, you could obtain the following sums:

Sum	Keys
1	1
2	2
3	2+1
4	4
5	?
6	4+2
7	4+2+1

Clearly, it is not possible to form 5 unless Y=4 (1+4) or Y=2 (2+1+2). Can you write a program to solve this problem in a reasonable amount of time?

Good and Bad Juice

Being a thrifty person, Jason bought a cheap orange juice, but after drinking one glassful, he decided it was just too horrible to drink. So he bought some better juice and drank one glassful from the container which he then filled up with cheap juice. If he continued this process—drink a glass and top up with cheap juice—what was the ratio of good juice to bad in his last glass? Each container holds two quarts, and his glass holds eight ounces.

Longest Ladder

What is the longest ladder a painter can carry in a horizontal position around a corner in Genoa where a 4-meter wide alley meets one that is 2.5 meters wide?

The Rule of 72

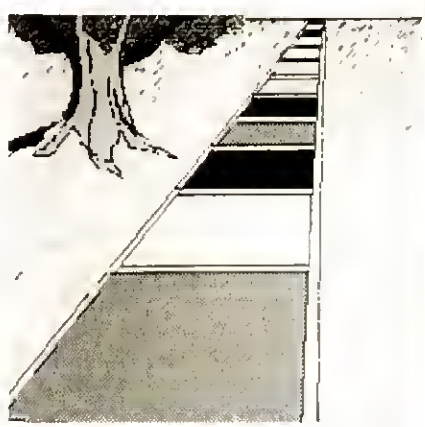
How long will it take you to double your money at a given rate of interest? To get an idea, you can use the "Rule of 72": simply divide 72 by the interest rate you are earning and the answer will be the approximate number of years. Can you explain why this works?

Lucky 6

In how many different ways can the digits 1 to 6 be arranged following the pattern:

1 2 3 4 5 6
1 2 3 4 6 5
1 2 3 5 4 6
1 2 3 5 6 4

If you continue this sequence, what will the sixth number be? The 60th? And, if it exists, the 600th?



Flagstone Walk

A man has red, gray, and blue flagstones for making a walk (one stone wide). How should he lay the stones so that no pattern is immediately repeated, that is, no consecutive stones are the same color, no consecutive pairs of stones have colors in the same order, no three stones have the same sequence as the preceding three, and so on. What pattern should he use and what is the maximum length it can be?

The Okidata Microline 292 with optional cut sheet feeder.

Microline 292 Printer

Okidata's newest impact printer offers reliability, versatility, and speed

By BETSY STAPLES

Price: $699
Interface: Parallel (IBM or Microline); RS-232C serial
Speed: 240 cps/draft; 200 cps/utility; 100 cps/NLQ
Summary: A reliable dot matrix printer with loads of print options.
Manufacturer:
Okidata
532 Fellowship Rd.
Mt. Laurel, NJ 08054
(609) 235-2600
(800) OKIDATA

The Okidata Microline 292 is a versatile dot matrix printer. Okidata calls it a "personal printer," but just what makes it personal is not clear, other than that it probably was not designed for the very heavy duty use it might get in some offices. I have been giving it moderately heavy use in my office for well over a month now, however, and it has never complained or showed any sign of fatigue.

With dimensions of 15.4″ × 10.8″ × 3.2″, the Microline 292 is on the small side of average in size. And it weighs in at a dainty 12.5 lbs., so you don't have to think twice about rearranging your work area or taking the 292 upstairs to the other computer.

The platen knob and power switch are on the right-hand side of the printer, where they are very convenient for right-handed computerists. The cable connects on the left-hand side, where it cannot interfere with paper feed in any way—an excellent design.

The control panel on the front of the printer has Power, Select, and Alarm lights and Line Feed, Form Feed, TOF (top of form) Set, and Select buttons, all of which perform the expected functions.

They also perform some very unexpected functions. For it is this set of buttons that invokes the wide range of print options that give the Microline 292 its versatility.

Print Options

To enter the Menu Select mode, from which you have access to these options, the manual advises that you can either turn the machine on while holding down the Form Feed button or, if the printer is already on, press the Form Feed and Line Feed buttons simultaneously. When you do, the printer responds with a MENU PRINT? message and you are ready to go.

Once you are in the Menu Select mode, you can step through the menu of print options by pressing various combinations of buttons. Among the options available are print mode (utility, NLQ, or draft); characters per inch (10, 12, or 17); emphasized, enhanced, double width, double height, and italic type styles; proportional spacing; character sets for foreign languages; page length; and quite a few others that will be of interest primarily to those who need to print graphics or modify the functions of the printer for specialized tasks. A sample options menu appears in Figure 1.

Each of the options is well documented in the Reference Guide, which also includes a table of possible combinations. You can, for example, combine NLQ mode with double width characters but not with double height characters. You can print in italics if you are using NLQ or utility mode but not if you want the extra speed of draft mode.

If you want to print only selected parts of your document in a given fancy typestyle, you must have a printer driver for an IBM-compatible printer. The 292 works perfectly well with the Epson-compatible SMM804 driver that comes with *1st Word*, as long as you don't want to italicize the name of a magazine or print the name of your company in boldface within a page of normal text.

I did not have an IBM-compatible printer driver, but a representative of Okidata assured me that one exists for the ST and that it allows you to make use of all the special features, including printing of bit image graphics. Be sure to ask for a copy of the driver when you

buy the printer.

Print Quality

The Microline 292 incorporates a "dual-nine-pin" printhead that permits single-pass printing even in NLQ mode. The printhead contains two columns of nine pins offset by one half of the pin diameter to allow overlapping of dots.

The quality of the NLQ mode is quite acceptable. It will not fool your friends into thinking that you have sprung for a daisywheel, but it should serve well for most business and personal documents. And 100 cps is very impressive for NLQ printing.

In utility mode, print quality is, again, acceptable. It is instantly recognizable as dot matrix output and is, in my opinion, slightly less attractive than what I have come to think of as the "standard" Epson-compatible typeface; many of the characters, particularly capital letters, seem quite "dotty." But at 200 cps, I don't mind putting up with a little dottiness.

Draft mode printing is somewhat smaller than the other modes and considerably less attractive. I haven't found any use for it, myself, but if speed is your object, you may be willing to sacrifice some print quality for that extra 40 cps.

Graphics can be printed in densities ranging from 60 × 72 dpi to 288 × 144 dpi in black and from 60 × 72 dpi to 144 × 144 dpi in color.

Paper Feed

No matter what print mode you use, you will have to cope with the somewhat primitive paper feed mechanism on the 292. Paper feeds in from the rear of the printer and around behind the platen, at which point you have to catch the holes of your pin feed paper on sprockets that flank the platen. This process certainly isn't complex, but I often had difficulty catching the right holes on the sprockets. Once threaded correctly, the paper feeds without any further problems.

The unfortunate thing about the pin feed mechanism of the 292 is that it does not allow for printing on labels or other smaller than 8½″-wide forms. The sprockets can be moved out about ½″ on either end of the platen to accommodate paper up to 10″ wide (including holes), but, because they are essentially extensions of the platen, they cannot be moved in for narrower jobs.

An optional tractor feed option kit with acoustic cover is available for the 292. Designed for large-volume printing, it adjusts for paper and forms of between 3″ and 10½″, but it will cost you another $45.

Friction feed for single sheets works like a charm; the paper feeds in easily and is held tightly all the way the bottom of the page. An optional cut sheet feeder, which handles up to 170 single sheets at a time, sells for $299.

Another nice feature of the 292 is a paper-out sensor that allows you to print to within ¾″ of the bottom of the page. If you need to get closer to the bottom edge of the paper, you can override the sensor from the print options menu.

Documentation

The Microline 292 comes with two beautiful booklets to help you use it. The 26-page Printer Handbook leads you step-by-step from unpacking the printer to replacing a fuse. The Handbook is clearly written and profusely illustrated with excellent four-color photographs.

The Reference Guide is 108-pages long and provides much more technical information, including a lengthy section on programming the printer from Basic. The Guide is nicely typeset in two colors on heavy paper with lots of charts and sample printouts. The entire documentation package is very professionally presented and leaves virtually nothing to the imagination.

I suspect that for most users the most useful section of the Guide will be the one that describes the Menu Select mode, including all the possible print options and combinations.

Of less interest to Atarians will be the specific instructions for using the 292 with *Wordstar 3.0*, *Lotus 1-2-3*, and *Supercalc 3* for the IBM.

When you purchase the 292, you must specify what kind of computer you intend to use it with so that Okidata can provide the proper "personality module." The personality module, actually an interface, is a cartridge that plugs into the printer and into which your printer cable plugs. The personality module that allows the 292 to talk to the Atari ST is the "IBM Compatible" version, so all the documentation you receive assumes that you are using (Perish the thought!) an IBM or compatible computer.

The Microline 292 comes with font design and color screen dump software. Unfortunately, it too assumes IBM compatibility (it comes on a 5.25″ disk) and won't be of much use to Atarians.

The Bottom Line

The bottom line really is the bottom line. The Microline 292 is a very capable printer. Sturdy and reliable, it offers a host of print options that you may or may not need or be able to use, and it carries a rather hefty price tag of $699 ($589 for the printer and $110 for the personality module).

If you have some special printing needs that are not addressed by the less expensive plain vanilla—or even chocolate—printers currently on the market, you may want to take a closer look at the pralines and cream Microline 292. ■

```
Double width in NLQ mode.
Double height in utility mode.
This is a sample of 240 cps draft mode.
This is a sample of 100 cps NLQ mode.
This is a sample of NLQ mode, emphasized.
This is a sample of NLQ mode with proportional spacing.
This is NLQ mode without proportional spacing.
```

PRNT MODE	DRAFT
CPI	10
LPI	6
EMPHSZD	N
ENHNCD	N
DBL WDTH	N
DBL HGHT	N
SUP/SUB	SHRNK
ITALICS	N
CHR SET	SET1
PROP SPC	N
LANG SET	0
GRAPHICS UNIDR	N
PG LNGTH	11
COLOR	BLK RBN
SEL DLL CHR SET#	0
SKI OVER PERF	N
AUTO LF	N
PPR OUT OVRD	N
PRNT REG	0
FORM TEAR OFF	1-2/3"
OWS	N
AUTO LOAD	Y

Figure 1.

War Of The Words

A detailed, feature-by-feature comparison of two popular word processors for the ST—Regent Word and Word Writer ST

A bit of background by way of introduction. As I was embarking on a book about the origins of the nuclear arms race in certain events of the late 1940s, I decided that it was time I acquired a computer as an aid in both writing and research.

My fantasy was—still is, come to think of it—that it would make work go much more smoothly if I had a word processing program that would automatically provide me with all the historian's scholarly apparatus. This would then leave me free to concentrate on content and style while the computer worried about arranging things so that just the right amount of space was reserved to accommodate my notes at the bottom of the page and a long single-spaced and indented quotation did not interfere with the automatic pagination of my otherwise double-spaced text.

The recommendations of friends and *Consumer Reports* led me to the Atari 1040ST, a computer that I concluded would do everything I wanted at a price I could almost afford. That was the easy part. What came next was another chapter in my never-ending education, as I searched for the software that would translate my fantasy into reality.

To make a long story short, I ultimately had the good fortune to stumble over *Word Writer ST* from Timeworks, a program that doesn't promise the moon but does carry out the elementary chores of word processing reliably, intelligently and with reasonable dispatch.

When I proposed a review of it to *Atari Explorer*, I was asked if I could also comment on *Regent Word II* from Regent Software in the same article. Still searching for a program that will meet my somewhat specialized needs, I was delighted to comply. With that bit of background out of the way, let's have a comparative look at the programs in question.

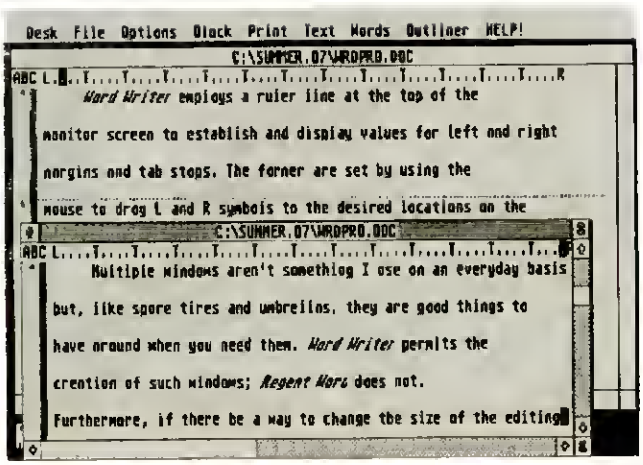
Manuals and Disks

Although *Word Writer* and *Regent Word II* are similar in several respects, when it comes to documentation, they could hardly be further apart. When you purchase the former, you receive a marvelously complete, well-illustrated, and lucid 142-page User's Manual printed in three colors, tab-indexed and bound in a sturdy three-ring looseleaf binder. If there is a better set of instructions for an ST program written, I certainly am unaware of it; the printer tutorial alone puts this one in a class by itself.

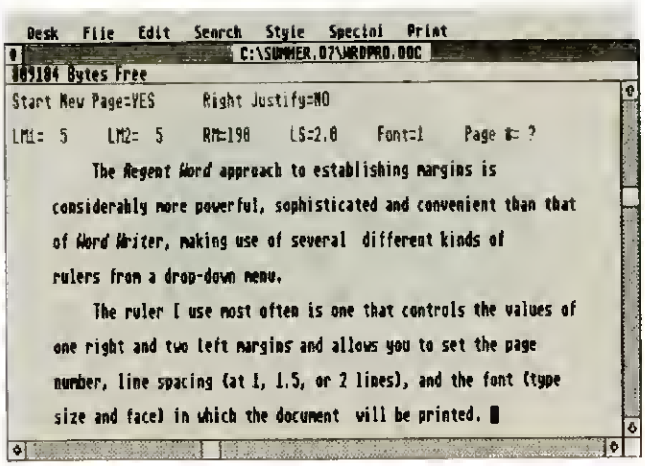
I only wish that more software publishers would meet the high standard for user's manuals set by this Timeworks product. I would gladly pay a few dollars more for a set of instructions that anticipated—and answered—all of my questions about a program.

The manual for *Regent Word* hardly merits comparison in the same breath. It is less than half as long as the *Word Writer* manual, poorly printed (in black and white) and held together with staples. Worse still, it doesn't begin to do

By FRANK KOFSKY



Word Writer employs a ruler line at the top of the screen and permits the creation of multiple windows.



Regent Word displays margin settings and other important information at the top of the page.

justice to the program it purports to describe.

There is no mention in its pages of, for instance, a drop-down Rulers menu or keyboard commands that delete and insert lines, turn on different type faces, etc. Some of this information is available from the window opened by the Help key, which, by the way, is also ignored by the manual.

A similar disparity exists between the disks you get with each of the programs. *Word Writer* is not copy-protected, so you can make backup disks and working copies that eliminate the files you are unlikely to need, thus reserving more of your disk for storage.

The *Regent Word* disk, in contrast, is copy-protected, and the only way to get a backup is by sending another $10 the manufacturer. As far as I am concerned, this policy simply adds $10 to the purchase price.

Bombs and Crashes

Regent Word can be made to malfunction and grind to a halt in two fashions. One is by positioning the cursor where it will cause a drop-down menu to open while the program is loading; another is by striking any key while a menu is open and then selecting an item from the menu.

What happens in both cases is that the program hangs up, and you must reboot to regain control. This, of course, destroys any work in progress that has not been saved.

I have yet to discover a way to cause a similar crash in *Word Writer*—thank heaven.

***Word Writer** uses an outstanding and logically consistent system to place special type faces on the screen.*

File Handling

If I had to summarize the difference between these two programs in a single sentence, I would say that where *Word Writer* includes any number of features designed to make it as convenient as possible to use, *Regent Word* inclines more to a bare-bones approach. This divergence is certainly evident in the way each program treats files.

In the *Regent Word* drop-down File menu, you find only four options: save a document, load a document, clear a document from the window, and quit. *Word Writer*, at the other extreme, nearly overwhelms you with options, letting you load a document (Open), save a document under its current name (Save), save it under another name (Save as), save it and go on to something else (Save and Abandon), combine it with another document (Merge), throw it away in disgust (Abandon), or Quit.

The file menu in *Word Writer* also offers a very handy Erase option, selection of which calls up a file management box that lists every file and folder on the disk, including the backup files automatically created whenever a document is either saved with the same name more than once or printed.

I presume that the authors of *Word Writer* felt that if the program was going to engage in such wholesale file-saving, the least they could do for the user was to allow him to unclutter his disk without having to return to the Desktop. I concur, heartily.

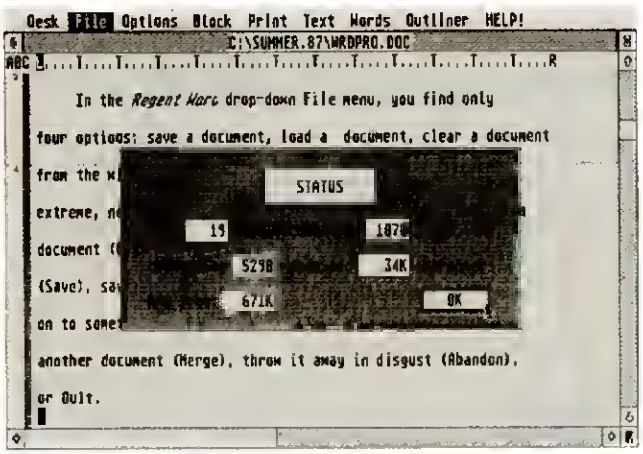
Multiple windows aren't something I use on an everyday basis, but, like spare tires and umbrellas, they are good things to have around when you need them. *Word Writer* permits the creation of such windows; *Regent Word* does not. Furthermore, if there be a way to change the size of the editing window in *Regent Word*—a virtual necessity when working with multiple windows—I have yet to make its acquaintance.

Margins, Justification, and Spacing

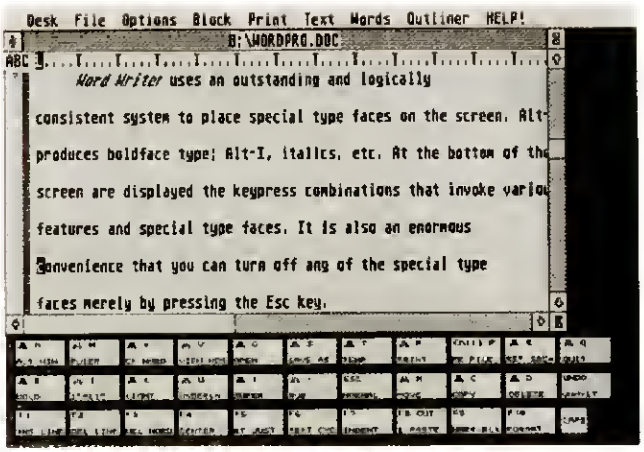
Word Writer employs a ruler line at the top of the monitor screen to establish and display values for left and right margins and tab stops. The former are set by using the mouse to drag L and R symbols to the desired locations on the ruler line; the latter can be inserted or deleted by clicking anywhere along the ruler line.

By comparison with the rest of *Word Writer*, this system is somewhat clumsy—especially so where the right margin is concerned. This margin cannot be set at an initial value greater than 71 until some text has been written on the screen.

In partial compensation, *Word Writ-*



Word Writer has a convenient Status check option that displays information about the size of your document. A word count can also be obtained for any Regent Word document.



Word Writer keypress combinations for special type faces and functions are always available at the bottom of the screen.

er permits you to write with a left margin of 0 and then offset the text to the right during the printing process. Thus, you can have the final margins of your choice while still keeping as much as possible of your text visible on the screen as it is being created and edited.

Another drawback to the margin options of *Word Writer* concerns indentation. If, for example, you want your first line to be flush left and all subsequent lines in a paragraph indented a certain number of spaces, you must use the F7 function key to indent the second line. Thereafter, new lines will begin at the indented margin—if, that is, the new line is preceded by an automatic carriage return. If you press Return, the next line will begin not at the indented margin, but at the extended one determined by the position of the L symbol. And reformatting operations with paragraphs that have indented margins are anything but smooth.

Besides showing margin and tabulation settings, the ruler line in *Word Writer* also indicates the horizontal position of the cursor—a useful feature lacking in *Regent Word*. But *Regent Word* has an option that displays both page and line numbers, whereas *Word Writer* tells you only the number of the page on which the cursor is positioned. Here, as was frequently the case in writing this review, I found myself wishing that I could combine features from both programs.

The *Regent Word* approach to establishing margins is considerably more powerful, sophisticated and convenient than that of *Word Writer*, making use of several different kinds of rulers from a drop-down menu.

Word Writer ST

System: Atari ST
Price: $89.95
Summary: Excellent word processor, overflowing with features
Manufacturer:
Timeworks, Inc.
444 Lake Cook Rd.
Deerfield, IL 60015
(312) 948-9200

The ruler I use most often is one that controls the values of one right and two left margins and allows you to set the page number, line spacing (at 1, 1.5, or 2 lines), and the font (type size and face) in which the document will be printed.

To activate this feature, you select the Rulers menu, then choose a Margins ruler. (Other items on this menu are a ruler that allows you to create a page break and change right-justification, two rulers that can be used to define the contents of headers and footers, and the option of deleting any previously-inserted ruler.)

Clicking on the appropriate part of the ruler enables you to use the keyboard to set any of the foregoing variables. With this system, it is easy to establish a format in which the first line of a paragraph has a different length from that of subsequent lines, because you can set two left margins—L1, which applies to the first line of the paragraph only, and L2, which applies to the remainder.

Both programs have options for right-justification and double-spacing, and *Regent Word*, as noted above, also provides an intermediate 1.5 line spacing. Both allow you to change the line-spacing and/or justification of a block of text after it has been typed.

The main difference between *Word Writer* and *Regent Word* is that the former generally leaves less to the imagination by displaying the text with virtually all of the characteristics it will have when printed—a design I applaud.

Although I am partial to *Word Writer* for attempting to show the text on the screen as it will appear on paper, the ease with which *Regent Word* enables you to create and change primary and secondary margins, etc., may well be more important in the long run. On balance, therefore, I think that this aspect of the comparison favors *Regent Word*.

On-Screen Special Text

Word Writer uses an outstanding and logically consistent system to place special type faces on the screen. Alt-B produces boldface type; Alt-I, italics, etc. It is also an enormous convenience that you can turn off any of the special type faces merely by pressing the Esc key.

Word Writer also makes it more convenient than does *Regent Word* to use

sub- and superscripts, by permitting you to turn them on with Alt-+ for superscripts and Atl- - (minus) for subscripts and off with Esc—once more, a nice touch from *Word Writer*, especially in comparison with *Regent Word*, which controls subscripts from a drop-down menu.

On the negative side, *Word Writer*, unlike *Regent Word*, underlines the spaces between words as well as the words themselves. This is a definite drawback for someone who wants to use the more dramatic underlining, rather than the less dramatic italics, to call attention to the title of a book and journal, emphasize a word or phrase, etc.

A minor difference between the two programs appears in the systems for engaging special type faces. *Regent Word* uses the Control key plus one letter key for this function—Control-I for italics, etc. Unfortunately, it assigns the boldface command to Control-D instead of the more logical Control-B. If you press the latter, you end up at the Bottom of the document.

And, as noted above, *Regent Word* requires that you press two keys to move from a special type face back to the standard one.

Although the two programs are similar in most respects, I think there are enough points where *Word Writer* demonstrates an advantage over *Regent Word* to give the former an edge in this area. As usual, though, a combination of the best features of each would eclipse either.

Other Aspects of Text Appearance

I don't want to have to remember margin settings, tab positions, the line-spacing and type of justification I have selected, whether new text will be inserted into or written over old, the status of the Caps Lock function, etc., and *Word Writer* obliges me nicely in this regard by displaying the settings I have selected on the screen.

Regent Word, whose authors seem to incline toward the philosophy that the less shown on the screen the better, does not. The one exception to this generalization is that *Regent Word* lets you choose whether you wish to have the page-and-line location displayed. I do, and for the life of me, I can't imagine why *Word Writer*, ordinarily so generous in supplying information, displays page numbers but omits those for lines.

Here is another exception to the foregoing generalization—one that works in favor of *Word Writer*: that program hides embedded printer codes, whereas *Regent Word* leaves them showing on the screen. With *Word Writer*, after inserting a new line by pressing the F1 function key and then typing the code characters, pressing the F6 key causes both the code and the inserted line to disappear. A checkmark is placed to the left of the left margin to remind you of the existence of the code, and a click of the pointer on the checkmark (usually) makes the code reappear, giving you the opportunity to modify or delete it.

Printer codes in *Regent Word* are shown on the screen at all times, preceded by a symbol indicating what they are, then are dropped out during the printing process. The problem with this arrangement is that the program treats the codes as part of the text for the purpose of determining the length of the line and maintaining the format. As a result, if the codes cannot be positioned in empty lines above and below a paragraph, the lines containing them appear appreciably shorter when printed.

To select an international character set on an Epson LQ-800 printer, for example, can require a code segment nine characters long, and switching back to the standard character set takes another eight or nine characters. If both codes are on the same line, that line will print about 25 percent shorter than those to either side of it—not good.

There are, then, numerous minor differences in the ways in which *Word Writer* and *Regent Word* place text on the monitor screen. No one of these is overwhelming in and of itself, but taken cumulatively, they establish the clear superiority of *Word Writer* in text appearance.

Editing Operations

The most common block operations—moving, copying, and deleting passages of text—are handled almost identically by both programs, with the arrow keys and/or mouse used to move the cursor and various function keys used to define the beginning and end of the block.

For some reason, *Regent Word* has no separate command for deleting a block, using instead the Move command to remove a marked block from the text and place it into a copy buffer, where it remains until inserted in the document or overwritten by a new block.

Because block operations so often involve cursor movements, it makes sense to discuss them here. The two programs are, as usual, similar for the most part, with *Word Writer* having just enough additional versatility to make it the preferred choice. Both *Word Writer* and *Regent Word* allow you to move the cursor to either end of the current line by pressing the appropriate arrow key and one other key (Shift or Control); this same other key, in conjunction with the up or down arrow, also moves the cursor a fixed number of lines in the corresponding direction.

Word Writer also offers the option of moving to the first letter of the next or last word by pressing Control plus the right or left arrow key—not an outstanding feature, to be sure, but one that again illustrates the amount of thought that the authors of *Word Writer* have invested in the program.

Word Writer wins another clear-cut decision over *Regent Word* in search-and-replace operations and making insertions in the document. In both cases, what the former does that the latter does not is preserve type faces when changes are made.

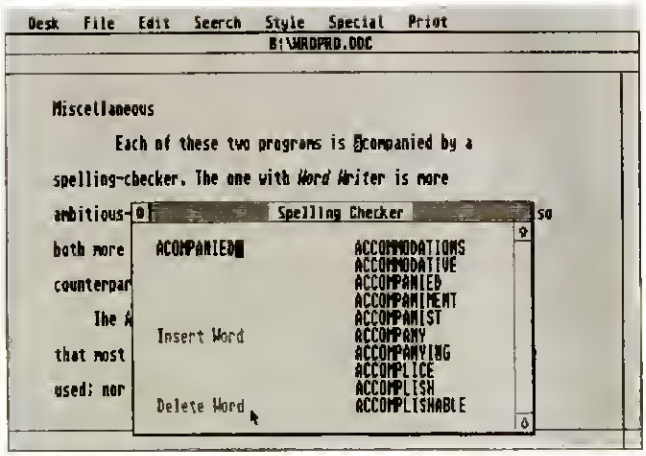
Suppose, by way of illustration, that you have written the boldfaced and italicized sentence, ***Actually, darling, I find you repulsive***, and you decide to insert ***utterly*** between the last two words. If you are using *Word Writer*, your inserted word will automatically be written in the same type face as those on either side of it.

What's more, if you then choose to use the search-and-replace function to

Regent Word II

System: Atari ST
Price: $79.95
Summary: No-frills word processor with some very powerful features
Manufacturer:
Regent Software
7131 Owensmouth
Suite 45A
Canoga Park, CA 91303
(818) 882-2800

Word Writer does not allow you to add words to its spelling-checker dictionary.



The Regent Word spelling-checker is included on the main disk and allows you to add words.

substitute ***consummately*** for ***utterly***, *Word Writer* will, again, respect your choice of type styles when carrying out the substitution. Bravo!

Faced with these same tasks, *Regent Word* will make its insertions and/or alterations in the standard type face, regardless of your preferences.

Word Writer has several other advantages over *Regent Word* when it comes to small-scale manipulation of text. If, for example, the cursor is placed on the first letter of a word and the F3 function key pressed, the entire word is deleted—a feature I have found much more useful than it sounds.

Word Writer also offers two other ways of deleting material within a single line. Pressing F2 anywhere in a line causes all of it to disappear, while pressing Control-Delete deletes all material from the cursor position to the end of the line. Deleting an entire line in *Regent Word* involves the additional step of moving the cursor to the beginning of the line before pressing Shift-Delete to make the line disappear. Pressing those two keys anywhere else in a line deletes to the end of the line.

Reformatting

If you have read advertisements for *Regent Word*, you probably know that ease of reformatting is the basis for many of the promotional claims made for it. And in truth, *Regent Word* is markedly superior to *Word Writer* in this regard.

The latter requires that, to reformat a paragraph after inserting and/or deleting words, you position the cursor at the head of the paragraph and press the F10 function key. All too often, however, this procedure has to be repeated several times to reformat the entire paragraph—and even then, it may fail to work.

Several times in the course of writing this review, I had to reformat paragraphs by deleting spaces to combine lines, pressing Return to end the line at or before the right margin, and so on. The erratic nature of its paragraph reformatting function is easily the gravest weakness of *Word Writer*, and it should be a matter of immense and immediate concern to the publisher.

Conversely, the fact that *Regent Word* automatically reformats paragraphs as you type is probably its single most outstanding feature—deservedly a selling point.

Word Writer includes an option that permits reformatting of the entire document in a single operation. In theory, therefore, you could sidestep paragraph reformatting difficulties by postponing the reformat until the document is complete. The trouble with this approach is that it forces you to work with lines that extend beyond the boundaries of the screen, necessitating constant shifting back and forth between the right and left extremes of the document to read what you have written.

Both word processors also permit you to carry out what I think of as after-the-fact reformatting, in which all or a portion of an already-written document is assigned a change in margins, line-spacing, or justification.

In summary, as far as reformatting is concerned, *Regent Word* gets the palm. It is clearly superior to *Word Writer*.

Printing

I use two criteria to evaluate how well a program works with respect to printing functions: does the program let me take full advantage of the capabilities of my printer in a straightforward way?; does it let me print my documents with exactly the appearance I wish? The printing aspects of *Word Writer* and *Regent Word* are of similar quality, with *Regent Word* holding a slight edge, based on the first criterion, *Word Writer* coming out ahead with respect to the second.

Both packages have ancillary programs for writing printer drivers to print italics, boldface, sub- and superscripts, etc., when given the appropriate command from the keyboard or menu). Both also allow you to *initialize* the printer driver so that it will cause the printer to produce a document in a given type size and face, e.g., double-strike elite, proportional pica, condensed double-width elite, and so on.

With *Word Writer*, I discovered that the most convenient way to print in these various sizes and faces was to create a different printer driver for each one and then select the one I wanted just before printing.

This procedure, however, was not quite free of bugs. For one thing, when a driver calls for the production of condensed double-width elite type, something in the *Word Writer* program causes the condensed feature to overwhelm that for double-width, resulting

in a document that is printed in condensed elite type. The only way I could discover to avoid this involved putting the condensed command in the driver, then embedding a printer code for double-width at the head of the document—not my idea of a fully satisfactory solution.

The *Regent Word* program for writing a printer driver is described on a half-page addendum to the user's manual—a state of affairs that did not fill me with confidence when I went to use it. But much to my delight, this program not only worked well but turned out to be a very sophisticated and convenient way of getting the most out of my printer.

In addition to establishing commands for the usual alternative type faces (bold, italics, etc.), the *Regent Word* program allows you to define ten different type size and style combinations, or fonts, such as condensed double-wide elite or proportional pica.

Thus, although both *Regent Word* and *Word Writer* can print documents in any of these fonts, the former goes well beyond the latter in allowing you to change from one to another within the same document by the simple expedient of selecting a margins ruler from a drop-down menu and clicking on the appropriate point within it—an admirable achievement. Timeworks programmers, take note.

It would be too much to expect, of course, that the same degree of thoughtfulness would characterize all of the *Regent Word* printing operations, and so it turns out. When it comes to letting you determine exactly how you want your printed document to look, *Word Writer* has the upper hand.

It allows you to determine the number of spaces between the top of the page and a header and between the bottom of the page and a footer; *Regent Word* does not. *Word Writer* has a simple embedded command to switch headers and footers off and on; *Regent Word* does not.

Miscellaneous

Each of these two programs is accompanied by a spelling-checker. The one with *Word Writer* is more ambitious—it can be used as or after you write—but it is also both more cumbersome and less useful than its *Regent Word* counterpart.

The *Word Writer* program exists on a separate disk that must be inserted in the main disk drive before it can be used; nor does it allow you to add words to its dictionary. Given that this dictionary lacks such relatively common terms as *prerogative*, *rejuvenation*, *archetype*, *subsequent*, and *Dixieland*, this last is indeed a grievous flaw.

Word Writer and *Regent Word* are both designed to dovetail with related business-oriented software manufactured by their respective publishers. Being pressed for time, not having these other programs on hand, and not being especially interested in the subject anyway, I am willing to take it on faith that *Word Writer* and *Regent Word* are relatively well-behaved and live up to the claims made for them in this regard, and let it go at that.

Regent Word allows you to define ten different type size and style combinations for printing.

Conclusions

When I began this review, I had no clear favorite in mind, no set of recommendations formulated. Having come now to the end, however, I believe a pattern has emerged. *Regent Word* can rightfully boast of some truly potent features: automatic reformatting; two left margins to permit a variety of paragraph layouts; rulers for ease in defining and changing margins, line spacing, justification, fonts, etc.; and multiple fonts immediately available within a single document.

Also to the credit of *Regent Word* are some less important but still useful features, including the display of line as well as page numbers, underlining that skips the spaces between words, and a spelling-checker to which new words can be added.

Yet on balance I find myself preferring to work with *Word Writer*. Why? As I see it, the choice comes down to this: with *Regent Word*, on the one hand, I have a very large convenience—automatic reformatting—and several minor but annoying inconveniences—switching off special typefaces, making insertions or search-replacements in alternate typefaces, cursor movement, and copy protection.

With *Word Writer*, on the other hand, I have one very large inconvenience—those same reformatting operations—but several very appealing amenities (including not only those just mentioned as lacking in *Regent Word*, but also the appearance of the text and the abundance of information on the screen, multiple windows, virtually complete control of how the document will be printed, relative immunity from crashing, and a user's manual second to none).

What is tolerable in the way of inconvenience lies, like beauty, in the eye of the beholder. I find it easier, in other words, to cope with the large but familiar, even predictable inconsistencies of the *Word Writer* formatting procedures than I do with the more diverse and more numerous shortcomings of *Regent Word*.

And inasmuch as all word processing operations are ultimately aimed at the production of a printed document, I have an especially difficult time overlooking the fact that *Regent Word* will not let me print what I have written exactly as I wish it to appear. Many other shortcomings in a program might be acceptable to me, but not that one.

What can you live with; what can you live without? Through writing this review, I have discovered that I can live with a word processing program that reformats text erratically—or without one that does so automatically—if it is elegant, simple and convenient enough in almost all other respects. On balance, therefore, I lean to *Word Writer* over *Regent Word*.

But it is axiomatic that not everyone will share what are, in the final analysis, my aesthetic preferences. For that reason, I have tried to set forth as explicitly as possible why I consider each program superior in certain areas. You may not agree with the amount of relative weight I have assigned to each of these areas, but I hope that, at the minimum, I have given you enough information to make an informed choice of your own, even if it be different from mine. ■

InSyst Inventory System

SoSoft offers an efficient program to help small businesses with inventory control

InSyst is a small business inventory system for all 8-bit Atari computers. Written in Action, this menu-driven system is fast and easy to use. One comprehensive screen permits you to add, edit, post, and inquire about inventory items, while another screen presents report options. The program is efficient, reliable, and effective *if*—and this is a big if—you can use the item format provided in the program.

Two versions of *InSyst* are provided on the distribution disk. On one side is version 2.23 for Atari 800 and XL computers, and on the other, version 3.0 for 130XE computers. Both versions are compatible with Atari DOS 2.0 and 2.5 and OSS DOS XL.

The main differences between the two systems involve item capacity (4000 in v3.0 vs 1700 in v2.23), disk drive requirements (v2.23 can run with one or two drives; 3.0 requires two), and overlay requirements (3.0 does not require overlays, because the entire system fits in one large program file.

Before even discussing how to create an inventory file, the manual spends several pages discussing daily backup procedures for your inventory data files. These sections clearly should be heeded if you plan to entrust the valuable records of your business to a few molecules of iron oxide on the surface of a disk.

Using the Program

From the Main Menu, you can choose to restart the program, exit to DOS, or select one of the three main functions of *InSyst*: edit or add records, produce a report, or do file operations (sort, load, create new file, etc.).

The screen you will use most is the Edit/Add screen (see Figure 1). From this screen you can add new records and view and edit existing records by scrolling through the file starting at any point you choose. The Edit/Add screen is the backbone of the system, and because it is fixed in its format, it must "work" with your business.

For each item, the system can store up to seven pieces of data. The item number can be up to 12 characters (alphanumeric or punctuation) long. The description can be up to 24 characters long. Other pieces of data you can capture include the retail price, physical location (bin, aisle, etc.), vendor number (0 to 99), and minimum and maximum stock levels.

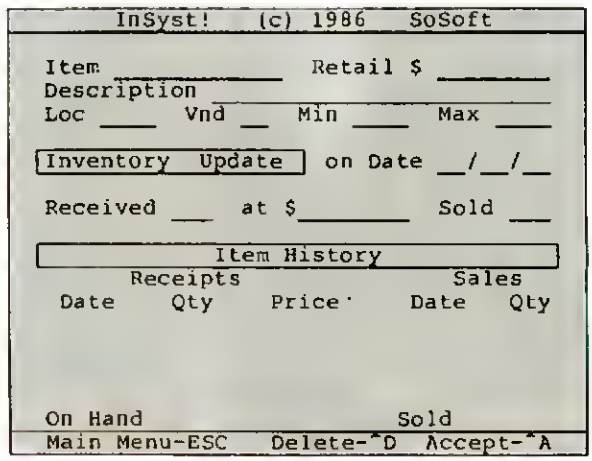


Figure 1. Edit/Add menu.

System: 48K Atari 8-bit computer
Price: $79.00
Summary: Efficient small business inventory program, if your business fits the format.
Manufacturer:
SoSoft
2513 #E Sylvester Rd.
Albany, GA 31705
(912) 888-0821

If this arrangement suits you, great, you'll love *InSyst*. On the other hand, if you want a flexible inventory data base program that can keep track of your business inventory as well as a software or book library (with alphabetic fields for title, author, publisher, and remarks, for example), this isn't it.

The Edit/Add screen also includes an inventory update area with space for date, quantity received, cost, and quantity sold. An item history area shows the four most recent transactions. Once your inventory file is created, it is this update area that you will be using most.

The program provides some error checking. You cannot, for example, enter a date greater than 12/31/99, nor can you enter a quantity of items sold that is greater than the number you have on hand. The system, incidentally, works on a FIFO (first in, first out) basis.

There are a few limitations that prevent companies, say, the size of GM, from using the program. For example, the maximum number of items in one transaction is limited to 255, and the maximum dollar value is $32,768.

Reports and Documentation

InSyst is considerably more flexible when it comes to the reports it can produce. You can choose which fields are to be printed and their order on the report. You can also analyze your inventory on many different criteria, including last sale date, on-hand quantity, items under minimum stock levels, bin location matches, and vendor code. Up to ten of these reports can be saved on a disk to be recalled for later use, so you need not redesign a report each time you need it.

InSyst comes with a comprehensive 40-page looseleaf manual and a set of labels for your main and backup disks. The program is provided on a non-protected disk, so you can easily make a backup. In case of trouble, SoSoft will provide telephone customer assistance—a generous offer in this impersonal age.

In our admittedly light use of the program, we had no problems that could not be solved by simply reading the manual, but it was nice to know that further help was just a phone call away.

If your small business can fit within the confines of the format imposed by the program, *InSyst* may be just the thing to help your 8-bit Atari bring order to your inventory. ■

By DAVID H. AHL

TechMate

Szabo Software offers a strong chessplaying algorithm, fine graphics, and an unobtrusive user interface

System: Atari ST (color or monochrome)
Price: $39.95
Summary: A challenging chess program with excellent graphics
Developer:
Szabo Software
P.O. Box 623
Borrego Springs, CA 92004
Distributor:
MichTron
576 S. Telegraph
Pontiac, MI 48053
(313) 334-5700

By LARRY TAPPER

Playing *TechMate* on the Atari ST really feels like playing a tournament chess game. You make moves with a hand-shaped cursor controlled by the mouse on a handsome and authentic-looking board display. To the right of the board is an uncluttered panel offering Play and Quit options, a move counter, and a pair of clocks positioned just as they would be in a tournament game.

Like a human opponent, *TechMate* parcels out its time according to the difficulty of the situation, stopping to ponder when it finds itself in a tight spot. You can feel the tension mount as the clocks tick away the allotted time.

The secret of the verisimilitude of this program seems to be nothing more or less than very solid handling of the basics. What you get is a challenging and congenial opponent, not a dazzling sideshow. This pleasing recipe has two basic ingredients: Barbara Young's excellent Atari graphics and Alex Szabo's chessplaying algorithm.

Although the Atari ST is a relatively new machine, the chessplaying component of *TechMate* represents years of programming work. The core algorithm was originally developed by Szabo as a Master's thesis in computer science.

Such thorough groundwork is really a prerequisite for this kind of software development; writing strong chess programs is a great deal more difficult than designing other types of computer games from scratch. In a chess game, there are zillions of future possibilities to consider—far more than the largest number-cruncher could handle by brute force alone—which is why AI experts have always been intrigued by the challenge of chess programs; chess programs must be smart as well as efficient.

Brute force does have its virtues, however. Back in the early days of computing when memory was in short supply, clever heuristic programming made it possible to create chess programs that would play a decent game. But raw computing power is the main reason that there are now several programs available commercially that play better than the majority of tournament chessplayers.

In the last decade, most of the serious development of chessplaying programs has been concentrated on dedicated microprocessors (for the broad consumer market) or on giant mainframes (for the AI theoreticians). The field has been dominated by a handful of conspicuous stars like the Spracklen programming team, Ken Thompson of Bell Labs, and chess masters Hans Berliner and Julio Kaplan.

Alex Szabo seems determined to join this select group, and Atari users are fortunate that he has chosen the ST as his proving ground. In the past, the chess programs available for personal computers have tended to be hand-medowns from other machine environments, but *TechMate* has been designed to take advantage of the outstanding graphics and user interface of the ST.

How Does It Play?

How well does *TechMate* play? The most pertinent answer seems to be: well enough. Like most commercial programs, *TechMate* allows you to control the level of playing strength of your computer opponent. You do this simply by setting the clock; the more time *TechMate* has to think, the better it plays. Meanwhile, you can control your own level by setting time limits for yourself on the separate player's clock. No matter what your ability, there is, in principle, some time handicap that will make you and *TechMate* dead even.

This approach seems to me more flexible, more entertaining, and less mysterious than the more commonly encountered menu offering several fixed levels of play. To advertise that a chess program has 26 levels of play doesn't mean very much; it is like saying that a thermostat has 26 different settings.

Unless you are a strong tournament player, you will probably find that *TechMate* can give you substantial time odds. A reasonable first experiment

TechMate graphics have been enhanced substantially since this photo of the game screen was taken, but the overall layout remains unchanged.

would be to give yourself a generous amount of time (say one hour) and leave *TechMate* in the default "blitz" setting (five minutes for the whole game).

The overall style of play of *TechMate* seems roughly comparable to that of other chessplaying programs. It rarely makes a gross blunder and punishes you swiftly for your mistakes. *TechMate* seems to be less comfortable in quiet maneuvering positions. It tends to make reckless pawn advances, so a skilled strategist can take advantage of its impatience.

Like most chess programs, *TechMate* behaves hyperactively in desperate positions, throwing away all its pieces to postpone the inevitable. But in general, the chess "ideas" presented in the program look sensible and natural.

Computer chess fanatics will have to wait for a more precise competitive evaluation. According to Szabo, version 1.1 of *TechMate* is deliberately designed not to be too intimidating to the average player. This contention is borne out by the program's exemplary sportsmanship; *TechMate* allows you to take back moves, doesn't think on your time, and uses a relatively limited opening "book" (a canned repertoire to save processing in the early stages of the game).

Since an expanded book and a constantly vigilant opponent would be fairly straightforward programming improvements, it is easy to believe that future versions will be much more formidable if the programmer chooses to make them so.

One area that does need work is the problem-solving mode; in version 1.1, setting up problem positions is a cumbersome process.

The good news is that active development is still going on, and Szabo promises that owners of version 1.1 will be able to buy future versions at a substantial discount.

TechMate shows ample evidence of loving care on the part of its designers; it is easy to use, attractively presented, well and honestly documented. Nor does it have any bugs or quirks that I could find.

Future revisions could well make *TechMate* one of the strongest and most theoretically interesting chessplaying programs on the market. But even as it stands today, *TechMate* is a clear winner for Atari gamesters. ■

Larry Tapper is a chess master and AI programmer.

8-BIT BONANZA

EDUCATION SPECIAL!!!

10 EDUCATIONAL TITLES FOR $25.00,

Here's all you need to do. Choose any of the starred (*) programs listed in this ad. Attach your list to the regular Bonanza order form and return as usual. (Please include 2 ALTERNATE CHOICES.)

SAVE $45.00

RECREATION

STAR RAIDERS II, THE ADVENTURE CONTINUES! NOW AVAILABLE

DX5084	Star Raiders II	disk	$16.95
CXL4013	Asteroids	cart	$ 6.95
CXL4004	Basketball	cart	$ 6.95
RX8030	E.T.	cart	$ 6.95
CXL4024	Galaxian	cart	$16.95
CXL4012	Missile Command	cart	$16.95
CXL4027	Qix	cart	$ 6.95
CXL4008	Space Invaders	cart	$ 6.95
CXL4011	Star Raiders	cart	$ 6.95
CXL4006	Super Breakout	cart	$ 6.95
CXL4025	Defender	cart	$19.95
RX8042	Realsports Tennis	cart	$16.95
RX8044	Joust	cart	$19.95
RX8049	Jungle Hunt	cart	$19.95
RX8048	Millipede	cart	$19.95
CXL4022	Pac Man	cart	$19.95
RX8045	Pengo	cart	$19.95
RX8033	Robotron	cart	$19.95
RX8026	Dig Dug	cart	$19.95
RX8043	Ms. Pac Man	cart	$19.95
RX8029	Realsports Football	cart	$16.95
KX7102	Arcade Champ Kit (includes PacMan, Qix cart holder and two joysticks)	cart	$24.95

MUSIC SOFTWARE

FOR GRADES 7 THROUGH AOULT (available on disk only)

DX5081	**Music Painter** Transcribe your favorite songs and save them for later use, or compose your own songs and build a library of personal hits.	$19.95
AX2020	Music I (Theory Lessons)	$24.95
AX2026	Music II (Advanced Theory)	$24.95
AED80048 *	Music I/Term Notations	$ 6.95
AED80049 *	Music II/Rhythm and Pitch	$ 6.95
AED80050 *	Music III/Scales and Chords	$ 6.95
APX20100 *	Advanced Music	$ 6.95

MATH AND ARITHMETIC

The Secret Formula Series teaches mathematical concepts by asking the player to create formulas that will duplicate a series of numbers generated by the computer

Available on disk only

AED80020 *	Secret Formula—Elementary (grade 4–6)	$ 6.95
AED80021 *	Secret Formula—Intermediate (grade 6–8)	$ 6.95
AED80022 *	Secret Formula—Advanced (grade 9–12)	$ 6.95
AED80057 *	Basic Arithmetic (elem)	$ 6.95
AED80060 *	Counting (K–grade 2)	$ 6.95
AED80055 *	Metric & Problem Solving (grade 3–7)	$ 6.95
AED80009 *	Math Facts and Games (elem)	$ 6.95
AED80011 *	Division Drill (grade 7–9)	$ 6.95

SPELLING, GRAMMAR, AND READING

CX4126	Speed Reading (grade 9–adult)	tape	$24.95
RX8059	Skywriter (grade 1–6)	cart	$17.95
AED80001 *	Spelling in Context/1 (grade 1)	disk	$ 6.95
AED80002 *	Spelling in Context/2 (grade 2)	disk	$ 6.95
AED80003 *	Spelling in Context/3 (grade 3)	disk	$ 6.95
AED80004 *	Spelling in Context/4 (grade 4)	disk	$ 6.95
AED80005 *	Spelling in Context/5 (grade 5)	disk	$ 6.95
AED80006 *	Spelling in Context/6 (grade 6)	disk	$ 6.95
AED80007 *	Spelling in Context/7 (grade 7)	disk	$ 6.95
AED80008 *	Spelling in Context/8 (grade 8)	disk	$ 6.95
AED80054 *	Prefixes (grade 3–6)	disk	$ 6.95
AED80059 *	Pre-Reading (K–grade 2)	disk	$ 6.95
AED80067 *	Spelling Bee (grade 3–7)	disk	$ 6.95
AED80069 *	Word Games (elem)	disk	$ 6.95
DX5050 *	Mickey, Great Outdoors (gr. 2–5)	disk	$ 6.95

PROGRAM LANGUAGES AND INSTRUCTIONS

CXL4018	Pilot Kit	cart	$39.95
AX2025	Microsoft Basic II	disk/cart	$29.95
CXL4003	Assembler/Editor	cart	$24.95
CX8121	Macro Assembler	disk	$24.95
CX4117	Invitation to Programming III	tape	$ 7.50
KX7099	BASIC Tutor (includes Invitation to Programming I & II and two books, Inside Atari BASIC and Programming Tips and Tricks)	tape	$19.95

SCIENCE AND SOCIAL STUDIES

The Atarilab series teaches science through experimentation.

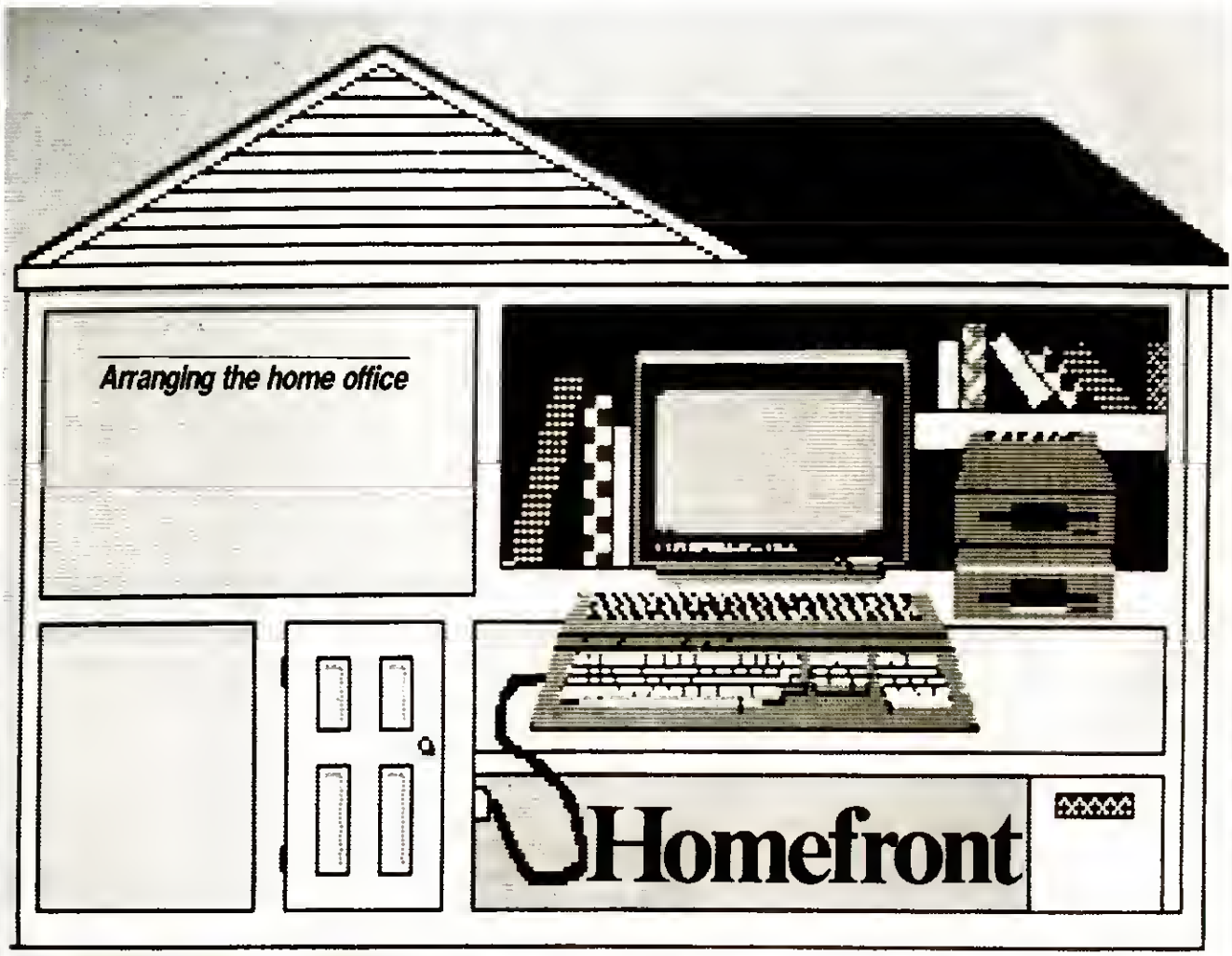
AED80013	Atarilab Starter Kit	cart	$59.95
AED80014	Atarilab Light Module	cart	$39.95
AED80017 *	U.S. Geography	disk	$ 6.95
AED80056 *	The Marketplace (grade 3–8)	disk	$ 6.95
AED80052 *	Earth Sciences (grade 7–9)	disk	$ 6.95
AED80053 *	Geography (grade 4–10)	disk	$ 6.95
AED80066 *	Expeditions (elem)	disk	$ 6.95
CX8106	Bond Analysis (adult)	disk	$ 9.95
CX8107	Stock Analysis (adult)	disk	$ 9.95
CX8108	Stock Charting (adult)	disk	$ 9.95

ATARI COMPUTER CONTROLLERS

CX21	Touch Pad Controllers (alpha numeric)	$ 9.95
CX30	Paddle Controllers	$ 9.95
CX40	Joystick	$ 6.95
CX75	Light Pen	$39.95
CX80	Trak Ball—HCD	$14.95
CX85	Numeric Keypad	$24.95

INK ROLLERS, RIBBONS, COMPUTER CABLES, AND ACCESSORIES

FC100626	Ink Rollers (NOW AVAILABLE)	$ 3.95
CO70263	Ribbon for XMM801	$12.95
CX87	Modem Cable for 850 Interface—for any modem	$24.95
CX853	16K RAM Module for 800 Computer	$19.95
CX82	Black & White Monitor Cable	$24.95



By ROXANE FARMANFARMAIAN

So you're open for business from home. The American dream is coming true in your garage, den, attic or barn. What about your Atari, though? Where have you put it now that it has become your chief cook and bottle washer, secretary, and administrative assistant?

Make no mistake, setting up a home office—physically—should be as serious a commitment as hanging out your shingle and opening your door for business. This is your world headquarters we're talking about. No kidding. This is where you will be spending hours at a stretch, working at the keyboard, sitting on a chair whose every contour you will come to know like the back of your, well, hand; shuffling papers or crunching numbers under a lamp that you will come to recognize like no other in terms of its shadows and glare.

Obviously, the environment in which you work must be conducive to building your business and making it a success. And that applies not just to you but to your computer too. If you mistreat it and it starts to misbehave, you and your shingle will be up a creek.

Stand back and take stock. Using the following checklist as a guide, assess how your current work setup rates in each of the areas that can most affect your health and output. If circumstances at home preclude a total solution, the accompanying tips are designed to suggest alternative solutions that might work instead.

Space

Have you set up your Atari so there is enough desk space around it for papers and other materials you need for your work? Or do you find that you use your lap as a secondary work surface, breaking your typing stride to glance down every few seconds?

Cramped work quarters lead to mess, and from mess to stress is a short step (even for clutter bugs like me). By using your lap or a sidetable (or cabinet top) as a compensatory surface, you set up a rhythm, usually subconsciously, of looking down sharply or twisting to the right or left that can cause back strain and neck tension.

Furthermore, too little space probably means that you have shoved your

Atari up against the wall, or worse, hemmed it in on all sides, restricting ventilation. If you have surrounded the machine so it has no room to breathe or if you have gotten into the bad habit of placing papers on top of the monitor, you risk overheating and eventual system failure.

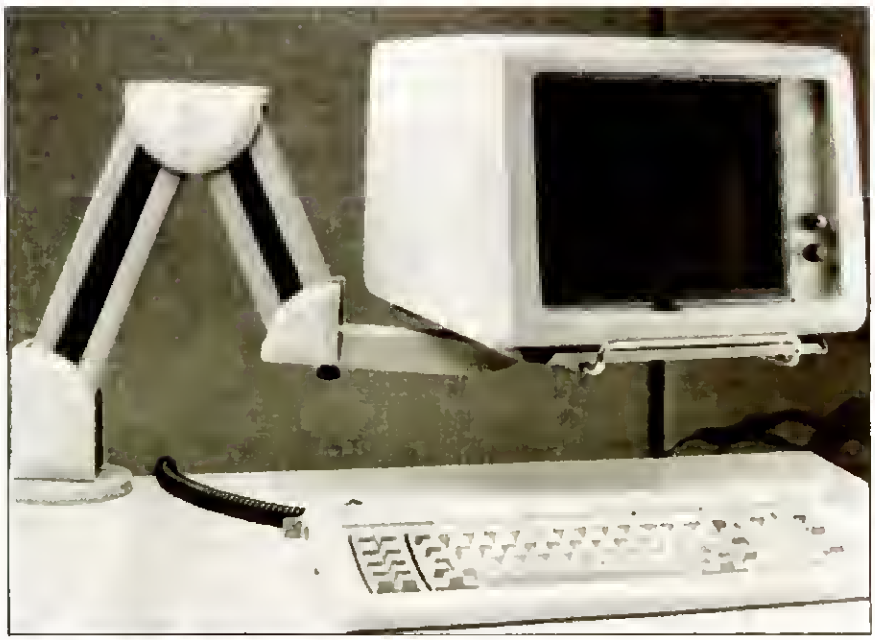
Tips: If you are not yet ready to exchange your house for a bigger, more "worker-friendly" model, there are some nifty ways of getting around the problem.

Very basically, what you need is a way to stow or stack your equipment so you can make better use of the space you have. One solution is a sturdy, freestanding tabletop frame (you can either buy or make one) on which you can stack your equipment vertically. This, in effect, places a shelf above your monitor—with ventilation space in between—where you can put your printer and modem, for example. Because the frame can stand a little way out from the wall, it provides a more or less hidden area in which to run your wiring that is still within easy reach.

Some commercially available computer furniture can achieve the same effect—though if you're going this route, check that the desk you buy can accomodate your wiring requirements and that it has room to store supplies (like disks and paper), too. Full-scale computer tables are usually extremely space efficient, enabling you to stack paper, printer, CPU, and CRT and still maintain an open work surface. Make sure the one you choose is sturdy, however; you don't want it to buckle under the weight of your equipment.

These computer desks can run anywhere from $150 to $400 depending on materials, type of construction, and design. Compro (P.O. Box 235, Londonderry, NH 03053, (800) 221-5399) sells one that is a bargain at $139.

Another solution is an extension arm that can hold either your whole computer or just your monitor. One end of this crane-like contraption (available primarily through mail-order houses) attaches to the table or the wall, and the other holds your equipment. When you need to work at the screen, you can swing the arm toward you; when you're done, you can fold it up and away against the wall, opening up the desk space in front of you. Commercial versions run about $100 depending on whether they hold the whole computer or just the monitor. The CRT Valet from USA-Flexruns (135 North Brandon Dr., Glendale Heights, IL 60139, (800) 872-3539), which sells for $98 is one example.



The CRT Valet by MicroComputer Accessories.

Computer console stowaways are a third space-saving possibility. These attach under your table, hiding (and protecting) your console until you need it. When you do, you can slide or pull the console out, and sometimes slightly up for easy use. Keyboard drawers run about $50 to $75; when ordering one through the mail, check to make sure it will fit the single unit keyboard/CPU of your Atari. Though most good ones are adjustable, you don't want to get stuck with one that isn't if you can avoid it.

And then, of course, there is the printer stand, which reduces the space your printer paper needs by stowing it under (or over) the printer. Printer stands are easy to make and cheap to buy at about $19 to $40. Daleo Electronics (P.O. Box 494, Franklin, OH 45005, (513) 435-1058) is one of many suppliers.

Seating

Is your chair soft and cushy, better suited to watching TV than working at a monitor? Or is it a straight-backed kitchen special, a little high (or low) and just a bit rickety? Studies by the National Institute of Occupational Safety and Health (NIOSH) are showing that bad seating, particularly over long periods of time, can cause back pain and even strain the muscles in your upper legs. The reason: a hunched, unsupported back puts an enormous amount of weight and pressure on the tailbone. This stresses the back muscles, and if the seat is soft, or too hard, the problem is intensified.

In an effort to compensate, you subconsciously start trying to support your back by tightening your leg muscles—an endeavor that is completely useless but that succeeds in wearing out and tensing the thighs.

A seat that is too high or too low will also cause leg strain as you constantly shift position, trying to get comfortable. Ideally, say the experts, you should have a chair that adjusts to your particular shape and size—with a back tilt, a seat that can be set higher or lower, and some sort of support for the small of the back. Dream on, right?

Tips: Lasso the best chair you can find without disrupting the general peace and harmony of your household. It's better to err on the hard side, since by adding a hard, flat pillow or two, you can usually compensate. There is really nothing you can do to firm up a soft chair. If the chairback is straight and uncomfortable, hang or tie a small round cushion to it so that the cushion presses gently against the small of your back. You'll be surprised how much it helps—even those who have never had a back twinge in their lives will notice the difference. This remarkable improvement is due partially to the tilt that such an arrangement gives the upper body, creating an angle, which, according to

the experts, is very, very important.

Though opinions have varied in the past, the general consensus today is: set yourself up so that you are looking down at the monitor. This relieves pressure on your neck and—you guessed it—your tailbone.

Once you start raking in the gold, consider spending some of it on a good office chair. No single purchase will give your home-business a more established feel (because a corporate chair really looks like a corporate chair). Such a chair will cost a small fortune ($100-$300), but it is money well spent on your physical well-being. And, after all, we apparently spend as much time sitting as we do sleeping—about half our lives.

Wiring

Is your floor covered with black spaghetti? Have you casually thrown your power supplies over the side of the table so they dangle, hidden perhaps, but dangerous? Negligence, when dealing with wiring is, very simply, courting disaster (a state of affairs your insurance agent won't take lightly). It also makes a very bad impression on visitors (let alone clients) upon whom you rely to recommend your services.

There are no shortcuts on this one. Get a power strip with enough outlets to accomodate your plugs. Lay your power supplies neatly on the floor, or on a low table or shelf. Tie up the unused lengths of wire with plastic garbage bag strips (the toothed, all-plastic strips are better than the paper ones with wires in them that may eventually cut through your cabling), or loop the wire over hooks in the wall.

Tidy arrangement of your cords and cables is the only way to ensure that you won't unwittingly disconnect the wrong device when you can least afford to lose a file, or that a springing cat won't yank your monitor onto the floor when playing swat and claw with the power supply.

Naturally, it's best if you draw power for your computer and peripherals through a single powerstrip. First, because powering all the equipment in your system from the same outlet eliminates the possibility of grounding and polarity incompatibilities, and second, because it simplifies the matter of protecting your computer and accessories from sudden power failures and other mischief.

Now that you are in business, the extra cost of a surge-protecting powerstrip is deductible—go for it. Once everything is plugged into the strip, clearly label the line that runs from the strip to the wall outlet—no electronic wizardry short of an "uninterruptible power supply," costing several hundred dollars, can save you if somebody searching for a place to plug in the vacuum cleaner pulls your main power cable out of the wall.

Labeling cords becomes even more important if your power strip will occasionally be used to hook up additional, temporary equipment. For example, one of the systems in my home occupies every plug in a large power strip, save one. The last remaining plug is used, alternately, for hooking up a music synthesizer and a modem. But what if, one day, I wanted to have both the synthesizer and the modem running at the same time? I'd have to unplug something else—maybe the printer—which would be risky if the system were turned on and the plugs not clearly labeled.

A space-efficient computer desk from Compro.

Storage

Are your disk files easy to access? Are the disks themselves easy to reach and logically labelled? How about documentation? Do you have the books you need at your fingertips? What about computer paper—is it where you can reach it? Is other stationery there too? Where do you keep blank disks?

More difficult to manage, but equally important, where do you stash the extra supplies and cables—like the null modem cable—that you use very rarely, but need absolutely when you do. Are your tools kept there too, or are they in the kitchen or the garage along with all your other tools? If your home office shares the computer with the kids, is their stuff mixed in with yours—or is it stored separately?

This is just a partial list. Computers breed "accessories" (what about printer ribbons and cleaning disks?), and as you may already have realized, so do home offices. Pandemonium will ensue if you don't get a grip right at the start.

Take stock of all the different categories of materials that you need to store and all the different types of storage media that you use (files, disks, log books, printouts, etc.). Don't underestimate what your situation will be six

***Now that you're** in business, the extra cost of a surge-protecting power strip is deductible—go for it.*

months or a year from now. Then set up a workable filing system and a logical storage system.

Get a second opinion on your setup—better to discuss possible flaws with someone else at this stage than to discover them yourself a year down the road when your habits have been set and change is time-consuming and probably even costly.

A filing cabinet of some sort is a must—you will be keeping records both for your clients and for the IRS (not to mention your accountant, lawyer, and insurance agent). So is a disk box, or two.

Tips: Filing cabinets don't have to be metal; cardboard ones that are cheap and much easier to move, even when full, are just as good and are available at Woolworths.

A set of plastic stacking kitchen bins (they come in great colors that will brighten your office) can work very well as storage for tools, supplies, and stationery. Plastic-covered wire mesh bins that hang underneath regular wall shelving can also provide extra space at low cost.

And just because you're working on your own at home doesn't mean you shouldn't get in-out trays and at least one standing file holder in which to store projects in progress.

The trick is to set aside separate cubbies and shelf space for every single one of your needs. This avoids the insidious growth of junk drawers and tabletop piles. Label as many storage areas as possible, and use different colors wherever you can for variety and for color association (bills in the red bin, new orders in the blue).

Finally, if business is booming but the mess is building up, consider hiring a temp for a morning to help you restore order. Since this is business, you may not be able to afford not to.

Lighting

Are you bothered by glare on the screen, so that you end up squinting a lot and tilting your head back or to the side? Is the lamp you work under shedding a harsh light on your work surface?

Lighting is a key factor in your work environment. Though it is often one of the most easily controlled, it is a frequent cause of stress.

Because the Atari monitor gives off its own light, you probably need less light in your electronic office than you would in an old-fashioned mechanical one. In fact, too much light can cause glare—and that applies to light pouring in from the outdoors as well as from an overhead lamp.

The best arrangement for computer work is a dim overhead lamp (or a shaded window) combined with a desk lamp that you can turn on and direct toward a particular task whenever you need it.

Tips: If the light in your work area is too bright, consider using a bulb with a lower wattage rating. Alternatively, get a different shade. If that doesn't work out, or if a conflict of interest makes it difficult to shade a window adequately to cut the effect of incoming sunshine, you can buy a filter to fit on your monitor to reduce screen glare (they run about $20-$40; the best are made with Polaroid materials). Turning on a desk lamp in broad daylight can help even out the light on your desk or other work surface.

The checklist could go on—I could look at static, security, even the traffic patterns of people in your home as they affect the placement of your office. These I leave up to you to consider on your own.

What is most important is to realize that each of these elements can have a significant impact on the success of your business. If you are cranky and achy from squinting at the screen, you are not going to make a good presentation, wow your next client with your accounting acumen, or design a stunning letterhead.

Likewise, if you can't put your finger on the name of a prospective client, your business will never fly—even if the service you could have rendered would have been superior to anyone else's, if only you could have rendered it.

The state of your office is in its own way as important as the state of your business. And don't forget that your Atari has to live in the office full time—even if you can escape and go "home" every once in a while. ■

A look at the Eastern Pennsylvania Atari Expo—now we know how it should be done

User Friendly

By BETSY STAPLES

I have been attending computer shows for close to a decade now, and I can honestly say that I have never seen one better organized or more carefully planned than the Eastern Pennsylvania Atari Expo that took place in Allentown, PA, this spring.

If your user group has not yet held an Expo, listen up, and I'll tell you what ABE'S ACEs did right—from an exhibitor's point of view anyway.

Without a doubt, the most unusual aspect of the show was the venue. The Eastern Pennsylvania Atari Expo was held in a very pleasant, pint-sized Hilton Hotel in the heart of downtown Allentown—only a mile or so from a major highway, surrounded by inexpensive parking facilities, an easy walk from reasonably priced restaurants.

What's so unusual about that? Nothing really, except that I had never before attended a show held in a hotel, and I liked the idea of being able to pop up to my room to change shoes or walk around the corner to get a decent meal.

I have exhibited at shows in convention centers, gymnasiums, cafeterias, and parking garages, but never have I been so *comfortable*. Usually, the show floor is blocks, if not miles, from the nearest hotel, and most are completely isolated, leaving attendees and exhibitors at the mercy of the "convention services" operator for all amenities. Have you ever had the misfortune to try one of their $4.50 hot dogs? Or how about the barbecued beef that is served with neither knife nor fork?

All this is by way saying that if you can find a nice hotel with a suitable room for your show, your exhibitors will love you, and some of the showgoers might be appreciative as well.

Shirley Bickert and Ralph Fenner, hard-working leaders of ABE's ACEs.

About Your Booth . . .

Another thing the Pennsylvania group did right was to include in the price of a booth everything most exhibitors need—no extra charges for furniture, draping, signs, or electricity. This is a much classier approach than presenting potential exhibitors with a laundry list of irritating extras.

And everything was in place when we arrived. I don't know what the situation was on Friday night when setup officially began, but when we arrived on Saturday morning, our booth was ready, club members were stationed around the area to provide help where needed, and all we had to do was open a few cartons and set out our wares. What a pleasure!

Shortly after we arrived, a large bouquet of helium-filled Atari balloons was delivered and tied to our booth, lending a festive air to the exhibit area.

The club had solicited door prizes from various vendors prior to the show, and Ralph Fenner, who was in charge of exhibit sales, periodically read the winning numbers over the public address system. His announcements were appropriate in number and content, if a tad loud.

How Public Is Your Public Address?

Which brings me to my next bit of advice: try to keep announcements to a minimum—in both number and volume. Don't turn the microphone over to the club clown for the weekend and assume that all 500 people in the room will enjoy his clever quips and Crazy Eddie imitation; more than likely, exhibitors and attendees alike will be longing to offer the fellow a microphone sandwich by the time Saturday lunch rolls around.

Take a lesson from Ralph: keep announcements short and concise and make them only when necessary. And do check the volume; most convention hall PA systems are capable of delivering many more db than the average Atarian's ears and brain need for accurate information processing. Remember, a whisper is a far better attention-getter than a shout.

Seeking Publicity

Some combination of ABE's ACEs and Atari's public relations agency managed to get some impressive media coverage of the Expo. There was at least one TV crew on hand Saturday afternoon, and the Sunday edition of the

Editors Staples and Jainschigg man the Atari Explorer booth.

local paper featured a story and photograph on the front page.

That's the way to do it, of course—arrange for the media coverage to run while the show is in progress. A story on page 6 of Monday's paper will make a nice keepsake and will net the club exactly zero dollars. So try to arrange for interested press people to visit the show early—maybe even Friday evening if you plan to begin setup then.

Regarding advance publicity, Sandi Austin says that posters have proved to be the most cost-effective form of advertisement. She recommends that clubs blanket the area from which they hope to draw attendees with large, professional-looking (try to avoid notebook paper and crayon) posters.

And don't overlook the potential of your own BBS and newsletter. The first people we talked to at the Allentown Expo were from Richmond, VA, and about three quarters of the subscribers we signed up at the show were from states other than Pennsylvania. How did they find out about an event so far from home? Probably—directly or indirectly—through a newsletter exchange.

So there you have a few hints and personal observations that I hope will make your Atari Expo more enjoyable and more profitable for all concerned. If those of you who have already held shows have additional tips, please write and let us know, we'll try to publish them here to help others. ■

You've Come A Long Way, Atari

You've come a long way, Atari, from the days when the only female folks at your user meetings were little girls whose regular babysitters were unavailable.

Now, Atari, you can look with pride at a groups like ABE's ACEs that has not only a substantial female constituency but a woman president.

Far from the stereotypical sexless nerd that was once thought to be the backbone of "computer clubs," Shirley Bickert is an attractive, efficient mother of three who has been involved with computers for only about two years.

She first became interested in learning to use the family Atari when her three-year old daughter wanted to use it. She asked her husband for guidance but found that taking computing lessons from one's spouse is only slightly less dangerous than taking driving lessons from one's spouse.

Not wishing to jeopardize her marriage, Shirley turned to ABE's ACEs. When she first started to attend meetings, she was one of only three women in the group, and she soon discovered that they were all there for the same reason; they were looking for a dispassionate third party to teach them what they needed to know about their computers.

Before long, Shirley had formed a Women's SIG with about two dozen members, and soon they were all hard at work learning the basics from other people's husbands. Since then, the Women's SIG has become the Beginners' SIG and fills, Shirley feels, a great need in the local Atari community.

Shirley has graduated from the Beginners' SIG and is now one of the few women presidents of Atari user groups. Her family's commitment to Atari hardware has grown as her involvement has increased, and the Bickert household now includes Atari 400, 600XL, 800, 1200XL, and 520ST computers.

We congratulate Shirley and all the other ABE's ACEs who worked to make the Eastern Pennsylvania Atari Expo a smashing success. ■

User Group Directory—Are You Listed?

You asked for it, but you're not going to get it unless you give us some help.

To date, only about 50 of the 500 or so user groups that Atari believes are in existence have responded to our call for information to be published in a User Group Directory.

We could just publish Atari's list of names and addresses, but we think it would be desirable to include additional information about each club. So we're going to give you one more chance.

Would those of you who have not already done so please designate one person in your user group to send us the following information about your group:

1. Full name and acronym.
2. Address.
3. Name of president.
4. Name of membership chairperson.
5. Frequency of meetings.
6. Location of meetings.
7. A phone number for prospective members to call for verification of meeting times/dates and other information.
8. Dues.
9. Does the group publish a newsletter? If so, give name and frequency.
10. Does the group maintain a BBS? If so, provide details.
11. Approximate number of members.
12. Anything else you think we should know.

Compile this information in legible form (please don't just send your newsletter) and mail it as soon as possible to:

User Friendly
Atari Explorer
7 Hilltop Rd.
Mendham, NJ 07945

This is your last chance to win fame, fortune, and new members. ■

Atari Fair Schedule

Rosemont, IL
SCAT, LCACE and CLAUG
July 25-26, 1987
Ramada Hotel O'Hare

Detroit, MI
MAGIC, Genesee Atari Group, ST User Group
August 29-30, 1987
Southfield Civic Center

Glendale, CA
ACENET
September 19-20, 1987
Glendale Civic Auditorium

Worcester, MA
Boston Computer Society Atari SIG
September 26-27, 1987 (tentative)
Worcester Centrum

Washington, DC
NOVATARI
October 17, 1987

Palm Beach, FL
Atari Club of the Palm Beaches
November 21-22, 1987

San Jose, CA
December 5-6, 1987

For further information, contact Sandi Austin, Atari Corp, 1196 Borregas Ave, Sunnyvale, CA 94086, (408) 745-2012.

Mark Williams C, Version 2.0

This new C compiler for the ST offers improvements on an already strong package

System: Atari ST
Price: $179.95
Summary: Improvements make this an excellent system for hobbyists and professionals alike.
Manufacturer:
Mark Williams Company
1430 W. Wrightwood
Chicago, IL 60614
(312) 472-6659
(800) MWC-1700

Version 1 of the Mark Williams C-language development system, has been a strong product since its first release last year. The massive package includes a high-quality preprocessor, compiler, assembler, and linker series, a disassembler, a debugger, a Unix-like command shell, and a MicroEmacs source text editor, along with a complete set of standard and ST-specific function libraries, an extensive Unix utility suite (diff, tail, etc.), an enormous amount of source code for editor, utilities, and example programs, and a manual close to 700 pages in length.

Yet, though feature-laden, the version 1 package did not contain everything needed to develop software on the ST. Though a full set of GEM functions was supported by the compiler itself, documentation of these functions was lacking. A Resource Construction Set—used to design pull-down menus, dialog boxes, and other GEM-related graphic structures—was also absent.

Ultimately, the purchase of Mark Williams C version 1 could be rationalized by professionals and serious hobbyists who had either already purchased the Atari developer's kit and utilities, or were willing to look far afield to find the missing documentation and software; but not by casual programmers or those merely looking to sharpen nascent C-programming skills.

Now, with the release of version 2 of Mark Williams C, most of the deficiencies of version 1 have been remedied. Mark Williams has taken a big step toward producing the compiler of choice for programming hobbyists as well as professionals.

Documentation

The new MWC manual is somewhat larger and more densely packed with information than the original. As with version 1, the bulk of the version 2 manual is occupied by a massive Lexicon—an alphabetically arranged guide to virtually every function supported by the compiler, shell environment, and utilities, plus a variety of useful information on C-programming concepts, code transportability, and other issues.

Complete GEM documentation is now included comprising a rundown of every major VDI and AES function, along with descriptions of correct calling procedures and often, detailed usage examples. In addition, all ST BIOS (Basic Input/Output System) calls, XBIOS (extended BIOS) calls, GEMDOS calls, and, to a lesser degree, Line-A calls, are documented.

This wealth of information should enable a programmer who has purchased Mark Williams C version 2 to develop a fairly comprehensive idea of how to program the ST with little in the way of additional reference material.

On the other hand, though most of what you need to know is represented in the Lexicon, finding it is sometimes not an easy task. Alphabetical ordering means that items are not grouped by function, and the index is not the "analytical" type used in most complex documents.

As a result, if you are stuck for a way to handle mouse input, you can't just look up the word *mouse*. Instead, you must start with at least a vague idea of how the mouse is handled by the ST operating system, then search the summary entries pertaining to each subunit of the OS: VDI, AES, TOS, BIOS, XBIOS, GEMDOS, etc., to find proper names for the functions you want to explore in greater detail.

Beginning ST programmers will require further references to develop a comprehensive overview of the system. Beginning C programmers will need at least one solid tutorial or reference work on standard C before they will be able to manage the Mark Williams documentation.

By ALEX LEAVENS

The Package

The MWC version 2 package consists of the manual and four single-sided disks, which contain the preprocessor, compiler, linker, assembler, disassembler, debugger, editor, utilities, libraries, a rebootable RAM disk program, and bonus source code.

Again, a Resource Construction Set has been left out of the package (Mark Williams is in the process of developing one), so programmers who want to exploit the full functionality of the ST in the easiest possible manner will have to obtain an RCS elsewhere for the time being. Except for that, however, it is difficult to imagine any basic software tool that has not been included.

The disks are not copy-protected, and a comprehensive install program is provided for establishing required directories on your working disk(s) and setting up the compiler and utilities on your system in usable form.

While lengthy—about a half-hour—the installation process is more or less foolproof, defaulting to a standard directory configuration that is well thought-out and eminently usable. Custom configurations can also be arranged.

To run Mark Williams C version 2, you need, at minimum, a 520ST and two single-sided disk drives or a 1040ST configured to emulate a two-drive system using the RAM disk utility included. Additional memory and drive capacity render the process of program development far more convenient, however. A hard disk, for example, is a real boon.

The Environment

The MWC programming environment is set up around a command-line interface "shell" called msh, which Unix fans may recognize as a combination of the popular Bourne and Berkeley Unix shells. Basically, msh replaces the GEM desktop, accepting commands from the keyboard to perform most disk operating system functions (change directory, catalog, etc.), execute programs, and generally manage the edit-compile-test-debug process.

Because this process consists largely of executing programs (such as cc, the compiler, or me, the editor) with passed arguments (including various function "switches," commands, filenames, etc.)—something a keyboard-oriented environment like msh is very good for—it can be argued that this kind of shell system is more convenient than GEM for the serious C programmer.

The other advantage of msh is that—as an environment optimized for use with Mark Williams C—the shell manages environmental variables, pipes, and other data structures that can be used to set up complex suites of cooperative programs in the shell environment.

Besides msh, the other corners of the MWC development tripod are me, the source code editor, and cc, the compiler. The editor is a full-blown implementation of MicroEmacs. Being command-driven (primarily by Control-key combinations), the editor is slightly user-hostile. However, it is fairly fast and reliable, permits editing more than one source file at a time, and is well integrated with msh and the compiler.

The Compiler

The Mark Williams compiler, cc, is the heart of the development system. Compilation under cc generally operates as a one-step process; the cc command calls the preprocessor, compiler, assembler/disassembler, and linker, as required.

The compiler suite is smart enough to recognize standard C-source, assembly language source, and object file extensions without special prompting, and can easily manage complex compile and link procedures involving multiple source files of different types. In combination with the included Make utility—which, when properly used, automatically keeps track of revisions to files involved in a project—the compilation process can be very simple indeed.

The shell, editor, and compiler are somewhat better integrated in version 2 of MWC than in the previous release. For example, if the compiler finds an error in your source code after having been given the directive -A for automatic editor invocation, the editor will be brought up, your source file loaded, an error message printed, and the cursor placed on the line containing the mistake. This level of automatism could eventually put us programmers out of a job.

Since the release of MWC version 1, the compiler itself has undergone some serious revision and improvement, too. An entire compilation pass has been eliminated, and the remaining passes (there are four, excluding the linker) have all been significantly enhanced, so much so that Mark Williams claims a 50% decrease in compile time over previous versions. While still not as fast as the Megamax C compiler, the new MWC is no slouch, especially when used with a RAM disk.

The compiler suite is smart enough to recognize standard C-source, assembly language source, and object file extensions without special prompting.

What is the cost of this increased speed? Mostly, the size of the code produced. Mark Williams C version 2.0 produces program files that can be about 15% larger than those produced by Alcyon, for example, though Mark Williams claims that version 2 programs will average about 10% smaller than those compiled under MWC version 1. An overall 20% increase in execution speed over version 1 is also claimed.

Mark Williams C version 2 clearly provides most of what constitutes a complete development environment for ST programming. Beginners will find the product tough going at first, though it should be added that Mark Williams provides very helpful and knowledgeable phone support to registered owners.

If you are willing to seek out suitable ancillary documentation and an RCS, MWC version 2 is a good buy for serious work on the ST.

Note: Owners of Mark Williams C version 1 may upgrade to version 2 for $50 plus shipping charges. Call or write the company for details. (And don't forget to tell them where you read about the product—*Ed.*) ■

Puzzles & Problems

Questions on page 29

Answers

Keyring Problem

We have written a horribly inefficient program (Listing 1) to print solutions (and near solutions, defined as arrangements that produce all sums from 1 to 19). One answer to the problem is 1-3-10-2-5. But there is another; can you find it?

You ought to be able to improve upon the program substantially. Array A represents each key. For ease in adding the values as we flip around the keyring, we use A(6), A(7), and A(8) to represent keys 1, 2, and 3. We put a 1 in each location in the S (sums) array for each sum that can be produced. What is your approach?

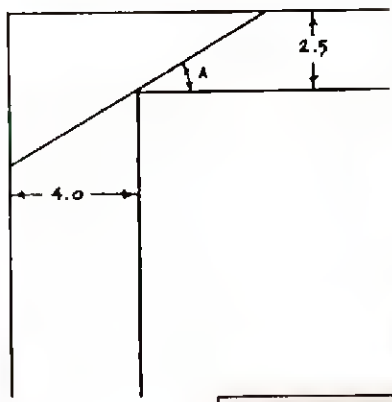
```
10 REM SOLVES KEYRING PROBLEM
20 DIM A(10),S(21)
30 A(1)=1:A(6)=1
40 FOR J=1 TO 17
50 A(2)=J:A(7)=J
60 FOR K=1 TO 18-J
70 A(3)=K:A(8)=K
80 FOR L=1 TO 19-J-K
90 A(4)=L:A(9)=L
100 FOR M=1 TO 20-J-K-L
110 A(5)=M
120 FOR P=1 TO 21:S(P)=0:NEXT P
130 FOR N=1 TO 5
140 S(A(N))=1
150 S(A(N)+A(N+1))=1
160 S(A(N)+A(N+1)+A(N+2))=1
170 S(A(N)+A(N+1)+A(N+2)+A(N+3)
    )=1
180 NEXT N
190 S(1+A(2)+A(3)+A(4)+A(5))=1
200 FOR P=2 TO 21
210 IF S(P)=0 THEN 240
220 NEXT P
230 PRINT "SOLUTION!"
240 IF P<20 THEN 250
245 PRINT P-1;" N  1 ";A(2);
    " ";A(3);" ";A(4);" ";A(5)
250 NEXT M:NEXT L:NEXT K:NEXT J
260 PRINT "THAT'S ALL FOLKS!"
```

Listing 1.

Longest Ladder

There are several ways of solving this problem, but one that lends itself to computer solution is to divide the length of the ladder into two segments—that in the 4-meter alley and that in the 2.5-meter alley. The length of each segment can be calculated by using sine or cosine (see diagram). The following program will compute all lengths between 1 and 89 degrees; you can refine the range as you get closer. By so doing, you will find the answer is 9.10977 meters.

```
10 DEG
20 FOR A=1 TO 89
30 L=4/COS(A)+2.5/SIN(A)
40 PRINT A,L
50 NEXT A
```



The Rule of 72

In the formula for compound rate of return, $P1=P*(1+R)^T$, substituting 2P for P1 and solving for T yields $T=LOG(2)/LOG(1+R)$. The program in Listing 2 compares the time to double your money at various interest rates using this formula with the "Rule of 72." The rule "works" because the log of 2 is approximately equal to .69 and, for small values of R, the log of 1+R is slightly less than R itself. Thus, if you try dividing R into .69, .70, and so on, you will find a good fit with .72 for interest rates between 4% and 18%, the range of most interest.

```
5 C=LOG(2)
10 FOR R=0.03 TO 0.2
     STEP 0.01
20 T=C/LOG(1+R):R1=R*100
30 PRINT R1;"   ";T;"   "
   ;72/R1
40 NEXT R
```

Listing 2.

Lucky 6

The digits 1 to 6 can be arranged in 720 different ways. We will publish the best reader-submitted program to show the entire sequence in a future issue. So crank up your computer and send us your program.

Flagstone Walk

We wrote a relatively simple program (Listing 4) that produced a sequence of 192 stones. We have no reason to believe that this is the maximum length. Can you modify our program or write one of your own to produce a rigorous solution? ■

Good and Bad Juice

The ratio of good to bad juice following the eighth and last transfer was 0.523. The program in Listing 3 prints the amount of good and bad juice in the container along with the ratio of good to bad.

```
10 G=64:B=0
20 FOR I=1 TO 8
30 G=0.875*G
40 B=0.875*B
50 B=B+8
60 PRINT B,G,G/B
70 NEXT I
```

Listing 3.

```
5 DIM I(400):S=3:T=1
10 I(1)=1:PRINT "1 ";
20 I(2)=2:PRINT "2 ";
30 FOR K=S TO 400:REM ADD FLAGSTONE
40 FOR M=T TO 3:REM SELECT COLOR
50 I(K)=M:REM SET COLOR
60 FOR N=1 TO K/2:REM TEST SEQUENCES
70 A=0:REM SET TEST INDICATOR TO BAD
80 FOR L=K TO K-N+1 STEP -1
90 IF I(L)<>I(L-N) THEN A=1
100 NEXT L
110 IF A=0 THEN 140:REM GET NEW COLOR
120 NEXT N
130 PRINT I(K);" ";:GOTO 160
140 T=1:NEXT M
150 PRINT :PRINT "TOTAL=";K;:GOTO 170
160 NEXT K
170 FOR J=K TO 3 STEP -1:REM GO BACK
180 IF I(J)<3 THEN 200
190 NEXT J:PRINT "ALL DONE?":STOP
200 S=J:T=I(J)+1
210 PRINT "  BACK UP TO ";S:GOTO 30
```

Listing 4.

Getting Started With Your Atari

Typing in programs from books and magazines is a great way to learn about your Atari and get some free software to boot—here's how!

By JOHN JAINSCHIGG

Those of you who arrived a little late for the Personal Computer Revolution probably don't remember how it all began. Back in the mid-1970's, inexpensive disk drives were still a gleam in Steve Wozniak's eye and the only practical ways that early computer hobbyists and primordial hackers had to exchange computer software were via paper tape and, well . . . via paper.

A visionary named David H. Ahl had just begun producing the world's first personal computing magazine—*Creative Computing*—and the program listings in its pages were, in effect, the first personal computer applications to be distributed on a mass scale. Those lucky, talented, or wealthy enough to have access to a personal computer with a keyboard and a Basic interpreter/editor (instead of just a big box with switches on the front) could actually type those programs in and sometimes even make them work!

Typing in programs from computer magazines thus has a long and honorable history. It is still probably the best way to become familiar with your system—its keyboard, editing features, controls, and peripherals; learn something about programming; and access, without charge, a vast and ever-growing library of quality software. Most of this software is written in Basic—an easy-to-use, powerful programming language originally invented for use by students at Dartmouth College and now the most widely-used language in personal computing.

Some Facts About Basic

Basic, like all programming languages, provides means for a sequence of instructions, comprising a program, to be delivered to, understood, and obeyed by a computer. In Listing 1 you see a very short Basic program that will run on any Atari 8-bit computer—400, 800, 600XL, 800XL, 65XE, or 130XE—under standard Atari Basic.

While instructions written in the Basic *language* may seem cryptic, they are actually very simple when expressed in English. The following is an English translation of the program in Listing 1:

1. Blank the screen.
2. Print the message HELLO, HOW ARE YOU? on the top line of the screen.
3. Wait until the user types something and presses the Return key.
4. If what the user has typed is the word FINE, then print the message THAT'S GOOD! on the screen.

By carefully tying together many, many instructions at this very simple level, a Basic programmer can create a complex, highly interactive, useful, and entertaining program. To assist the programmer in this creative task, Basic provides:

- A way of communicating with the computer, entering Basic commands from the keyboard, and correcting typing errors.
- Help in finding and correcting simple errors—*automatic error checking.*
- A way of testing to see if a given instruction will do what you hope it will—*immediate command execution.*
- A way of storing the list of instructions you type in, and keeping this list in proper order as you make changes, additions, and deletions—a *memory storage area.*
- A way of executing the whole list of instructions on receipt of a single command—program *interpretation* and *execution.*
- Various ways of tracking down and eliminating additional errors in a program—*tracing functions, run-time error checking.*
- Ways to save your program on some permanent medium—paper, tape, and/or magnetic disk.

If you review the features of Basic listed above, you will realize something quite interesting: Basic has most of the features of a typical application pro-

```
10 DIM R$(25)
20 PRINT CHR$(125); "HELLO, HOW ARE YOU";
30 INPUT R$
40 IF R$="FINE" THEN PRINT "THAT'S GOOD!"
```

Listing 1.

gram—a word processor, for example. When you are using a word processing program, you have a way of entering data (in this case, text) and commands into the computer; some kind of error-checking that tells you if you have pressed a wrong key or made some other mistake; a way of issuing single commands (like: "change all the 'recieves' to 'receives' in this document"); a way of storing your text (data) in memory—keeping it all immediately accessible even though there may be too much to fit on the screen at once; a way of tracking down further errors in your data—automatic spelling checkers and the like; and the usual save, load, and print commands that let you copy your data (text) from memory onto some permanent medium.

In fact, Basic is an application program—plain and simple. It's just an application program that understands a larger and more fundamental list of commands than your average word processor. So varied are the commands understood by Basic that it is, in fact, possible to compose sophisticated applications with them—applications like word processors and arcade games—that differ from their grown-up cousins only in that they run inside the environment provided by Basic, instead of the larger environment provided by the computer hardware.

In effect, Basic acts as intermediary—an "interpreter"—between programs written in its own command language—the Basic language—and the larger hardware environment in which Basic itself runs.

Getting Started

The steps involved in typing in a Basic program and getting it to run are as follows:

1. Get Basic fired up and ready to go.
2. Type in the program, using the Basic typing, editing, and error-checking features to correct your mistakes.
3. Save the program on some permanent medium—like a disk—so your work doesn't go to waste if something goes very wrong (not very likely).
4. Tell the computer to run the Basic program.
5. Correct any further errors with the help of the error-checking and editing features of the language, which are always available—one of the nicest things about Basic.
6. Save a final copy of the corrected program on a permanent medium.

Once you have typed in a Basic program, gotten it to work correctly, and saved a final copy, you can use the program any time you like by simply:

1. Getting Basic fired up and ready.
2. Telling the computer to load in the program from tape or disk.
3. Telling the computer to run the program.

We'll cover each of these steps in detail for both 8-bit and ST computers.

Getting Basic Fired Up

Atari 8-Bit Computers

Before you start: If you own an Atari 810 or 1050 disk drive, you will probably want to use it to save your Basic programs on magnetic disks. To do so, you must make certain that your drive or drives are properly hooked up, that you have made a working copy of your DOS disk, and that you have one or more formatted disks on hand. Read the introductory sections of your disk drive and/or DOS manual for instructions.

Basic was supplied in cartridge form with the original Atari 400 and 800 series computers and the 1200XL. If you have one of these computers and don't own an Atari Basic cartridge, you will need to buy one before you can run Basic programs on your machine. Send a check or money order for $15.00 plus $2.00 for shipping along with a letter requesting CXL4002, Atari Basic Revision C to Atari Customer Relations, P.O. Box 61657, Sunnyvale, CA 94088).

Owners of Atari 800XL and 600XL computers, which were sold with Basic Revision B, can obtain Revision C free of charge, provided the machine is still under warranty, by sending a dated proof of purchase to Atari Customer Relations at the address shown above.

To install Basic on an Atari 400 or 800:

1. Make sure the power to the computer is turned off.
2. Open the cartridge bay behind the keyboard by pressing the catch forward. Make sure your hand is in place to prevent the hood from flying up too quickly.
3. Hold the cartridge with the slotted end (the business end) down and label facing you. Insert it in the slot (the left slot on an Atari 800), and press down carefully until it is firmly seated. The cartridge should not wiggle loosely when correctly installed.
4. Close the cartridge bay.

Newer Atari 8-bit computers—the 600XL, 800XL, 65XE, and 130XE—come with Atari Basic built in, so no special installation procedure is required. If you turn one of these machines on without holding down the OPTION key, its built-in, "invisible Basic cartridge" is automatically activated. From this point on, startup procedures will be the same, regardless of which Atari 8-bit computer you own.

If you intend to use a disk drive to save your Basic programs, you should turn on the drive at this point and insert a *boot disk*—a formatted disk with the files DOS.SYS and DUP.SYS copied onto it via the H option on the DOS menu. If you have no such disk available or do not understand these terms, *do not* use the original DOS disk that came with your disk drive. Stop here and check your disk drive manual for instructions on how to create a boot disk.

Now turn your monitor or TV on, and switch on your computer (remember *not* to hold down the OPTION key if you are using an Atari 600XL, 800XL, 65XE, or 130XE). If you are using a disk drive, you will the see the drive become active for a few seconds and hear a regular beeping noise from the display (assuming the sound is turned up).

Otherwise, you will simply hear a gravelly sound, lasting a second or so. Then, the screen will turn light blue and the word READY will appear in the upper left-hand corner of your screen. Basic is now fired up and ready to go—on to step 2!

Atari ST Computers

Before you start: *Boot* your system just as if you were preparing to run a regular application program. If your ST is an early model that does not have TOS in ROM, start with whatever system disk you normally use. Owners of later model STs may boot from the disk drive or not, as they prefer, depending on whether they wish to have desk accessories present.

For the time being, leave the system in whatever resolution it is in when you normally start up. Make sure you have at least one formatted disk ready for saving Basic programs. If you do not know how to format a disk, check your ST Operator's Guide for instructions.

You should also be familiar with basic GEM operations: pointing and clicking with the mouse, moving and resizing windows, and selecting from menus. The Operator's Guide contains a tutorial that should get you on your way.

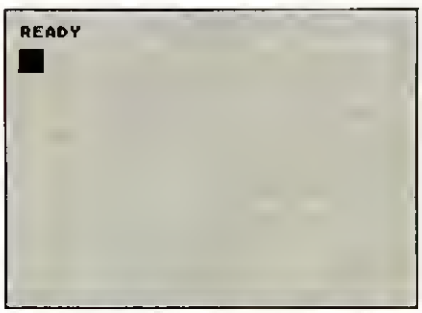


Figure 1a. Atari 8-bit Basic startup screen.

ST Basic is supplied on the Language Disk that came with your ST. However, a new, updated version of Basic has been prepared and will probably be ready for distribution by the time you read this; check with Atari Customer Relations at the address listed above for more information.

To get Basic fired up and ready, place the Language Disk in your main disk drive, Drive A. Double-click on the Drive A icon to open a window on the drive and display the directory. Find the icon labeled BASIC.PRG and double-click on it. The screen will clear, and, in a few seconds, the ST Basic desktop will appear; it consists of four windows, labeled LIST, OUTPUT, COMMAND, and EDIT (which lies on the desktop under the other three). The one you will use now is the Command window at the bottom of the screen, which contains the Ok prompt.

Now you're ready to go.

Compatibility

Now that you have Basic fired up, it's time to start typing.

But you can't just type in any Basic program you run across and expect it to work. A Basic program designed for an Atari 8-bit computer, for example, probably won't work on an ST, nor the reverse. Even though both machines are said to "run Basic," the versions of Basic that run on each machine, and the hardware of the machines themselves, differ sufficiently to make most of any Basic program written for one machine incomprehensible to the other.

This incompatibility extends to Basic programs written for other brands of computer, as well. Though it is possible to write "generic" Basic programs that can run on a wide variety of machines under an assortment of different versions of Basic, this is seldom done in practice.

Further considerations must also be taken into account. For example, though almost any program written for an Atari 800XL will run on an Atari 130XE, the reverse is sometimes not the case. The 130XE has more memory, and programs written to make use of this memory may not run on less powerful systems.

The configuration of your system, and the peripherals you own may also be significant in determining what programs you can type in. A graphics program that makes use of an Atari SMM804 printer, for example, may not be useful to you if you own an NEC printer.

To avoid disappointment, therefore, you must be careful to type in only those programs designed to run on your computer brand, model, and particular configuration, under the version of Basic that you own. This is seldom difficult to determine.

Computer magazines usually identify their programs clearly—more often than not in such a way that even someone who has not read the article or instructions accompanying a program listing will not be fooled into typing in a program that cannot run on his machine. *Atari Explorer*, for example, prints an "Atari Key" above each program listing, providing this information in concise and consistent format. Nevertheless, it is good practice to make a habit of reading all the material accompanying a program listing before trying to type it in.

When considering whether to type in a program, ask yourself the following questions:

- Was this program written for my brand and model of computer?
- If not, do I have reason to believe it will work on my machine? While differing in such details as amount of RAM memory available, all Atari 8-bit machines—from the 400/800 to the 130XE—are fundamentally compatible with one another. Likewise, though the 1040ST has twice the memory of the 520ST, the machines are internally very similar. What you need to ask is, does this program address aspects of a particular machine that don't exist in my own computer? If you are not familiar with programming, this is sometimes difficult to determine. If you're not sure, try another program.
- Do I have the version of Basic that this program requires?
- Does my system include all the peripherals this program requires? If not, are the peripherals I lack considered optional? For example, a program may be capable of using some peripheral—say, a printer—if it is available, but the basic function of the program may not be compromised if that peripheral is not available.

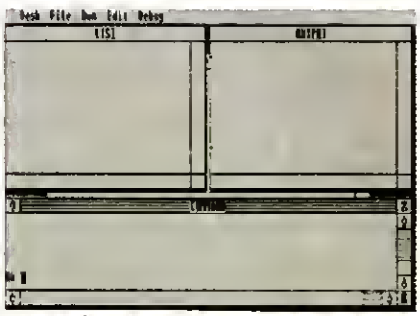


Figure 1b. Atari ST Basic startup screen.

Two short example programs—one for Atari 8-bit computers, one for STs—have been included in Listings 2 and 3 at the end of this article. These can be used for typing practice in the discussion that follows.

Program Typing

For most people, typing is an arduous, error-prone, and time-consuming task—even with a computer. Before beginning, make sure you are seated comfortably, can reach the computer keyboard without strain, and have enough light to see clearly what you are doing.

Now look at the computer screen. If you have followed the instructions given above for getting Basic up and running, your screen should look something like Figure 1a or 1b.

On Atari 8-bit machines, the word READY indicates that Basic is patiently waiting for you to type something in; the word Ok in the lower left-hand corner of the ST Basic screen means the same thing. Beneath this *prompt* (or next to it, in the case of the ST) is a cursor—a marker that shows where the next letter you type will appear.

Try pressing the A key. See what happens? The letter A appears (in lower case on the ST, unless you have pressed

the Caps Lock key), and the cursor moves one space to the right, marking the next position on the line.

Now let's get rid of that letter. To do this, 8-bit owners should press the Delete key, ST owners the Backspace key (there is a Delete key on the ST, too, but it performs a different function from that on the Atari 8-bit machines.) The letter disappears, and the cursor moves one space to the left, resting once again near the margin.

It's very much like using a word processor. You type something in, and if you don't like what you have typed, you erase it by using the Delete or Backspace key. There are other, more sophisticated ways of correcting mistakes, as well, but all you need to know to get started is here.

Now let's get a little more adventuresome. Press the A key again, to put the letter back on the screen. Then press the Return key.

Whoa! Something happened, right? What happened was that when you pressed Return, Basic assumed you wanted to register what you had typed—either as an immediate command, or as part of a program. But the letter A isn't one of the commands that Basic understands, nor does it conform to the special grammatical rules that would identify it as part of a program. So Basic complained, telling you that it didn't understand what you were trying to do.

Basic takes an interest in what you have typed only after you press the Return key, and not before. To test this, try typing in the entire alphabet, several times. When the cursor reaches the right margin of the screen (or the text-entry window, on the ST), just keep on typing letters—a few dozen more. On Atari 8-bit systems, the cursor will "wrap around" to the left margin, moving down to the next line, automatically. On STs, the text-entry window will scroll left, letting you enter more text than will fit on a single screen line. Now press Return. See? Basic didn't complain until you pressed the Return key, even though you filled up a complete line of the screen with nonsense.

Now, let's try giving Basic something it understands. Type in the following expression, exactly as it appears, and press Return when you are done:

```
PRINT "HELLO THERE!"
```

What happened? Your computer printed the message HELLO THERE! on the screen—it obeyed your command to PRINT what you told it to. On Atari 8-bit systems, the message is printed right underneath the line on which the command was typed. On ST systems, the message appeared in the OUTPUT window at the upper right.

Now let's try something a little more advanced. Type in the following, including the preceding number and separating space, and press Return when you are done:

Typing
in programs from computer magazines has a long and honorable history.

```
10 PRINT "HELLO THERE!"
```

Now what happens? Nothing at all, if you have followed instructions correctly. The number 10 tells Basic that you want to register what you have typed as a program line—part of a program that you are typing in—so it doesn't execute it right away. Instead, it stores what you have typed, including the number, in its program storage memory, and awaits additional instructions.

Let's give it some. Type RUN, and press Return.

What happened? Probably the same thing that happened up above, when we gave Basic an immediate command. The message HELLO THERE! appeared on the screen or in the OUTPUT window. By typing RUN and pressing Return, you told the computer to execute the program it was holding in memory—and it did just that.

Now let's try another command. Type LIST and press Return. The computer responds by displaying a *list* of the program statements it is holding in memory. Atari 8-bit systems print this list right underneath the line on which you entered the command. ST systems—somewhat more sophisticated—print it in a List Window at the upper left of the screen, but the result is the same.

Naturally, there is only one statement in this *listing*—the one you just typed in. Let's add another. Type:

```
20 PRINT "GOODBYE!"
```

Did you remember to press Return? Now, let's try executing the program by typing RUN and pressing the you-know-what key. What happens? If everything is in order, Basic will have printed HELLO THERE! and GOODBYE! on the screen.

Obviously, the new statement you just typed in was registered as part of the program. Now tell the computer to list the program by typing LIST. This time, both program lines should be printed out.

Now, let's remove one of the lines. To do so, just type 10 and press Return.

Try typing LIST again. What do you see? By giving Basic a number that corresponds to the number that precedes one of the statements in a program, you eliminate that statement.

Now, let's add something new to our program. Type:

```
10 PRINT "HI THERE!"
```

and press Return. What do you think will happen? To find out what Basic has done, type LIST. If you have followed instructions, the following listing should appear:

```
10 PRINT "HI THERE!"
20 PRINT "GOODBYE!"
```

Basic has accepted a new line 10 and has automatically sorted it into its proper position in the program. If you type RUN, the statements in the program will be executed in the order of their line numbers, and the following output will appear on your screen:

```
HI THERE!
GOODBYE!
```

Now let's review what we have learned:

- Basic commands and program lines are given to the computer by typing them in at the keyboard.
- Simple typing errors can be fixed using the Delete or Backspace key to erase the error and then retyping it.
- Pressing the Return key registers what you have typed, causing Basic to respond to it. Basic does not respond until you press Return, and does not mind if what you type runs over the right margin of the screen.
- Once you press Return, Basic will inform you if what you have typed in

does not make sense.

•Your Atari understands a variety of Basic commands, including PRINT, RUN, and LIST.

•If you precede what you type with a number, Basic treats what you have entered as part of a program and stores it in memory. Programs are built up in this way: from many statements, entered in sequence, each beginning with a unique number. The lines of a program are automatically sorted and executed (when you type RUN) with reference to these numbers, regardless of the order in which they were entered.

•A program line is erased by typing in its line number and pressing Return. A program line can be modified by typing in its line number, followed by new text and a Return.

Getting Down to Business

Now that you know the basics of how to interact with Basic, it's time to start typing in a more substantial program, like one of the example programs below.

As you just learned, a Basic program is made up of a sequence of statements or program lines, each beginning with a unique *line number* and ending with a press of the Return key. Typing in a program is simply a matter of typing the printed listing into your machine, pressing Return at the end of each program line to register it and place it in memory.

Take a look at one of the sample listings. Note that certain numbered lines don't fit on a single physical line within the space allowed—they run over onto the physical line underneath. This is not an indication that you should automatically press Return, as you would on a typewriter, when you reach the right margin of the page—it is just the result of the magazine having to print the listing in a confined space.

When you reach such a point, just keep on typing, moving your eye down one line, until the remainder of the *logical program line* has been entered. Only then should you press Return. Conversely, certain program lines that take up only one physical line on the printed page may not fit onto one physical line on your screen. Again, just keep typing, letting the cursor *wrap around* (or letting the window scroll left) until you have entered the entire line—then register it with Basic by pressing Return.

Obviously, the format of program lines—at least as far as line breaks are concerned—isn't very important. The content is. Basic is a very dense and precise language in which virtually every letter, number, space, and punctuation mark is significant.

Until you understand Basic well enough to know where the shortcuts lie, you should strive to duplicate the content of each line in the printed listing (including line number) exactly. Otherwise, there is a good chance that your program won't work. Copying a line exactly includes being careful to:

•Type capital and lower-case letters as indicated in the listing.

•Count spaces and other "filler" characters carefully, and copy them exactly where they appear.

•Distinguish between characters of similar appearance. Don't substitute a O (capital letter O) for 0 (numeral zero)—zeros are usually *slashed* to make it easier to identify them. Don't use an l (small letter L) for a 1 (numeral one). Don't substitute square brackets for parentheses, or apostrophes (') for quotation marks ("). While all these things may pass when preparing typewritten documents for human readers, your computer will be much less forgiving.

Note that certain program listings you come across may contain *special characters*—odd blocks, little heart-shaped characters, and other peculiar-looking things—or *inverse characters*—characters printed in a dark color on a light background. This is particularly true in Basic programs designed for Atari 8-bit computers, though ST listings may also contain characters that don't seem to be represented on the ST keyboard.

Instructions on what keys to press to produce these characters can be found in your computer or Basic manual. Magazines and books that print listings containing special characters also usually provide a key, giving instructions on how to produce them from the keyboard.

When typing, remember that if you make a mistake, you can simply use the Delete or Backspace key to erase the line back to the offending character and

TWINKLE, TWINKLE

ATARI KEY

- Any Atari 8-Bit Home Computer
- Atari Basic

```
10 DIM V1(42,2),V2(46,2)
20 PRINT CHR$(125):POSITION 5,10:POKE 752,1
30 PRINT "*TWINKLE, TWINKLE, LITTLE STAR*"
40 FOR I=1 TO 42:READ A,B:V1(I,1)=A:V1(I,2)=B:NEXT I
50 FOR I=1 TO 46:READ A,B:V2(I,1)=A:V2(I,2)=B:NEXT I
60 T1=0:T2=0:P1=0:P2=0
70 IF T1=0 THEN SOUND 0,0,0,0:P1=P1+1:SOUND 0,V1(P1,1)
,10,10:T1=V1(P1,2)*20
80 IF T2=0 THEN SOUND 1,0,0,0:P2=P2+1:SOUND 1,V2(P2,1)
,10,10:T2=V2(P2,2)*20
90 T1=T1-1:T2=T2-1
100 IF T1>0 OR P2<46 THEN 70
110 POKE 752,0
1000 DATA 60,1,60,1,40,1,40,1,35,1,35,1,40,2
1020 DATA 45,1,45,1,47,1,47,1,53,1,53,1,60,2
1030 DATA 40,1,40,1,45,1,45,1,47,1,47,1,53,2
1040 DATA 40,1,40,1,45,1,45,1,47,1,47,1,53,2
1050 DATA 60,1,60,1,40,1,40,1,35,1,35,1,40,2
1060 DATA 45,1,45,1,47,1,47,1,53,1,53,1,60,2
1070 DATA 60,1,81,1,47,1,60,1,45,1,60,1,47,1,60,1
1080 DATA 53,1,64,1,60,1,72,1,91,1,81,1,121,2
1090 DATA 47,1,81,1,53,1,81,1,60,1,81,1,64,1,81,1
1100 DATA 47,1,81,1,53,1,81,1,60,1,81,1,64,1,81,1
1110 DATA 60,1,81,1,47,1,60,1,45,1,60,1,47,1,60,1
1120 DATA 53,1,64,1,60,1,72,1,91,1,81,1,121,2
```

start typing again from that point.

Until you press Return to register the line with Basic, you can make all the changes you like in this way. Once you have pressed Return, Basic will check the line for additional errors. If it finds any, it will tell you about them. To correct errors in a line you have already registered, type it in again from the beginning, including the line number, and press Return to substitute the correct line for the incorrect one in memory.

There are also other, more efficient, ways to *edit* parts of a program listing, which you can check out in your computer or Basic manual. Note that the automatic error checking feature of Basic can spot only such errors as render a command or program line grammatically incorrect.

Basic cannot check for underlying flaws in logic or other errors that, while incorrect in the context of a particular program, are still legal in terms of Basic grammar. In other words, although error-checking can help you to some extent, you must still be very careful when typing in a program.

Insurance

Once you have typed in a whole program, the next step is to save your work so that if something goes wrong, it won't be lost. The Basic command SAVE lets you do this. Atari 8-bit owners who want to save a program on disk should follow this procedure:

1. Place a formatted disk (prepared in advance according to instructions in the disk drive manual—*do not* use your original DOS disk!—in your disk drive and close the drive door (or bring down the latch).

2. Type: SAVE "D:FILENAME", substituting a name appropriate for your program, eight characters or less in length, between the quotation marks where the word FILENAME appears, and press Return. The disk will whir for a moment, while Basic saves the program for you.

To reverse the process, bringing the program back into memory, type: LOAD "D:FILENAME", substituting the name you gave in the previous instruction for the word FILENAME, and press Return.

Owners of Atari 8-bit cassette systems follow a similar procedure to save and re-load their work. The Atari cassette drive must be properly attached, and an appropriate cassette inserted. To save a program, press down the Record and Play buttons of the tape drive simultaneously—the drive will not start moving when you do this, but will await a command from the computer. Type: SAVE "C:FILENAME", substituting a name of your choice, as above, and press Return. The computer will beep twice and write the program onto the cassette.

To reverse the process and re-load the program from the cassette into memory, start by re-winding the tape to where it was when you began. Then press down the Play button. Type: LOAD "C:FILENAME", substituting the name you gave in the previous instruction, and press Return. The computer will beep, and load the program from the cassette.

In general, cassettes are less reliable than disks, and it is a good idea to save each program twice, without rewinding the cassette in-between. That way, if one copy of the program turns out to be bad, you will still have another.

ST owners should follow a similar procedure to save their work:

1. Place a formatted disk (prepared in advance according to instructions in the ST Operator's Guide or ST Basic manual) in drive A.

2. Type: SAVE A:'FILENAME', substituting a name of your choice where the word FILENAME appears. Basic will save your program on the disk.

If you attempt to use the above command twice in a row (supposing you have already saved a program under a certain name, corrected an error, and are now trying to save it again using the same name) an error occurs. At this point, you have two options—save the new copy under a slightly different filename (many programmers save multiple versions of their programs under a related series of names, e.g. FILE1, FILE2, FILE3) or use the REPLACE command to replace the file containing the old version of your program with a file containing the new version, under the same name:

REPLACE A:'FILENAME'

TWINKLE, TWINKLE

ATARI KEY

- Any Atari ST Computer
- ST Basic

```
10 dim v1(42,3),v2(46,3)
20 sound 1,0,0,0,0:sound 2,0,0,0,0:sound 3,0,0,0,0
30 wave 7,7,0,0,0
40 fullw 2:clearw 2:gotoxy 21,7:print "*TWINKLE, TWINK
LE, LITTLE STAR*"
50 for i = 1 to 42:read v1(1,1),v1(i,2),v1(i,3):next i
60 for i=1 to 46:read v2(1,1),v2(i,2),v2(i,3):next i
70 t1 = 0:t2 = 0:p1 = 0:p2 = 0
80 if t1 = 0 then sound 1,0,0,0,0:p1 = p1+1:sound 1,10
,v1(p1,1),v1(p1,2),0:t1 = v1(p1,3) * 30
90 if t2 = 0 then sound 2,0,0,0,0:p2 = p2+1:sound 2,10
,v2(p2,1),v2(p2,2),0:t2 = v2(p2,3) * 30
100 t1 = t1-1:t2 = t2-1
110 if t1 > 0 or p2 < 46 then 80
120 sound 1,0,0,0,0:sound 2,0,0,0,0
1000 data 1,4,1,1,4,1,8,4,1,8,4,1,10,4,1,10,4,1,8,4,2
1010 data 6,4,1,6,4,1,5,4,1,5,4,1,3,4,1,3,4,1,1,4,2
1020 data 8,4,1,8,4,1,6,4,1,6,4,1,5,4,1,5,4,1,3,4,2
1030 data 8,4,1,8,4,1,6,4,1,6,4,1,5,4,1,5,4,1,3,4,2
1040 data 1,4,1,1,4,1,8,4,1,8,4,1,10,4,1,10,4,1,8,4,2
1050 data 6,4,1,6,4,1,5,4,1,5,4,1,3,4,1,3,4,1,1,4,2
1060 data 1,4,1,8,3,1,5,4,1,1,4,1,6,4,1,1,4,1,5,4,1,1,
4,1
1070 data 3,4,1,12,3,1,1,4,1,10,3,1,6,3,1,8,3,1,1,3,2
1080 data 5,4,1,8,3,1,3,4,1,8,3,1,1,4,1,8,3,1,12,3,1,8
,3,1
1090 data 5,4,1,8,3,1,3,4,1,8,3,1,1,4,1,8,3,1,12,3,1,8
,3,1
1100 data 1,4,1,8,3,1,5,4,1,1,4,1,6,4,1,1,4,1,5,4,1,1,
4,1
1110 data 3,4,1,12,3,1,1,4,1,10,3,1,6,3,1,8,3,1,1,3,2
```

Why save a program before testing to see if it works? Simple—if something goes wrong, you don't want to have to type the program in again from scratch.

Mistakes in a Basic program can cause strange things to happen; they can even, on occasion, cause whole programs to "self-destruct." If you have a copy on disk or tape, you have nothing to worry about—just LOAD it into memory, find your mistake, fix it, and save a new copy before trying to run it again.

Note that your valuable time is really the only thing at risk when working with Basic. The worst thing that can possibly happen (and even this is *highly* unlikely) is that something you do will accidentally erase a disk. Typing and programming errors cannot harm your computer, disk drive, or other peripherals in any way.

If something happens that makes you very uncomfortable (like your printer, all of a sudden, begins to spew forth reams of paper), don't panic. If worst comes to worst, you can just turn the system off and start up again.

Debugging

You have typed in your program and saved it on disk or tape. Now comes the moment you have been waiting for—time to test it out. As you probably remember, you can tell the computer to execute a Basic program that is in memory by typing RUN and pressing Return.

If you are a) an extraordinarily good typist or b) incredibly lucky, the program will work right off. The rest of us less talented souls who do not enjoy Divine protection will probably be greeted with a friendly error message or some other indication that the program contains one or more typographical mistakes. What to do now?

First, eliminate the obvious. Did you, by chance, type RNU instead of RUN? That will get you every time. No? Check the printed listing again. Did you accidentally type in a Commodore 64 program? Nope.

Once all the easy possibilities have been eliminated, it's time to look yourself squarely in the eye and admit that you might possibly have made one or two little typing errors. *Debugging* is the computer term for finding and eliminating these mistakes.

Until you have learned quite a bit more about Basic, you can't expect to be able to debug a program simply by examining its behavior. Instead, you will have to adopt the more systematic approach of checking each line you have typed against what appears in the magazine, searching for discrepancies between them.

To list a particular program line on your computer screen, line 40, for example, type LIST 40 and press Return. Line 40 should be displayed on-screen (it will appear in the List Window on the ST).

Complicated

mathematical equations with many levels of parentheses cause beginners more than a little frustration.

It is also possible to list larger chunks of a program to make checking go a little faster. To display all the program lines between, for example, line 40 and line 80, 8-bit owners should type LIST 40,80 and press Return. ST owners should type LIST 40-80 to accomplish the same thing. Note that Atari 8-bit Basic uses a comma to separate references to line numbers, while ST Basic uses a hyphen.

If you own a printer, checking your listing against the magazine is even easier. To get a printout of your work, turn on your printer, making sure that it is properly loaded with continuous-form paper. Eight-bit owners should type LIST "P:" and ST owners, LLIST, followed in either case by Return.

As you work your way through the program, checking for mistakes, keep in mind the need for absolute accuracy. Have you substituted a letter O for a zero? Dropped a few spaces between a pair of quotation marks? Put a comma where a period should be, or vice versa? Remember that the error you are looking for is probably quite subtle.

DATA statements—program lines beginning with the usual number, followed by the word DATA and a long list of numbers or words separated by commas, often cause beginning programmers a great deal of trouble. Because there is no visible pattern to most of the content of the line, typing it accurately becomes a very mechanical process—prone to error.

Complicated mathematical equations with many levels of parentheses also cause beginners more than a little frustration. Both DATA statements and math expressions tend to be very fragile, in the sense that one tiny mistake can change their meaning (and ruin your program) completely—check them carefully.

When you find your mistake, there are many ways of handling it using the editing features of Basic. The one surefire way, however, is simply to type the whole line in again from the beginning, including the line number, and press Return to replace the original line with the repaired one.

As you find and correct mistakes, remember to save a new copy of the program on disk or tape every so often. There is no sense in wasting the time you spend debugging any more than there is in wasting the time you spent typing.

Sometimes, you will be unable to find your mistake, even though you have searched through the listing over and over with all the diligence you can muster. Chances are still very good that the error is yours; only very rarely can problems with a program be attributed to errors in the original printed listing. *Explorer*, for example, tests its program both before and after publication, to ensure that errors have not crept in at any point in the makeup and printing cycle.

In the event that you cannot find any errors and the program still doesn't work as advertised, you have two choices. The first, and probably the best choice, is to seek the assistance of a more experienced Atari programmer. To find one, consult your local Atari user group.

Your second choice is to sit tight and wait for the magazine to discover its mistake and publish a correction—most will do so in the most timely manner possible.

On To the Wider World

You now know the basics of how to type in Basic programs. As you become more familiar with your Atari computer system, you will learn many additional techniques and tricks that will speed up the process of typing in and correcting published listings.

Careful study of the manuals that came with your Atari computer and peripherals will enhance your skills and understanding and ease the process of acquiring good, useful software from the printed page. ■

Software Survey

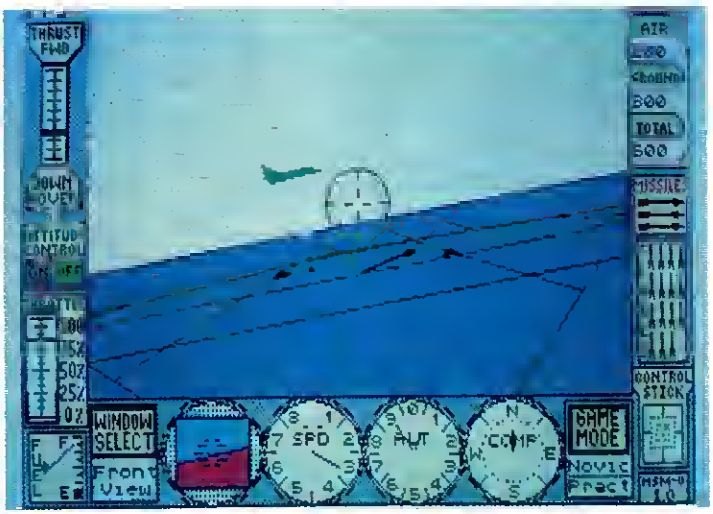
These short reviews will keep you up to date on some of the latest, greatest, and not-so-great software available for Atari computers.

If the play is truly the thing, *Arena* isn't likely to satisfy many of its customers. But if showy graphics that darn near look like video tape highlights are paramount, then *Arena* may be the game of the year.

Sporting the best graphics I have seen in a game program, *Arena* is 60% of a decathlon; six of those ten familiar events are included. On board are the 100 meters, long jump, shot put, javelin, high jump, and pole vault. A look at the missing events gives a good indication of the weaknesses of the program.

Gone are the hammer throw, two longer races, and the hurdles. Why the hurdles were left out is a mystery. But the reason for the omission of the other events isn't a case for Agatha Christie.

Participation in the six included events is accomplished by pounding a key on each side of the keyboard to correspond with the planting of alternate feet. The quicker these keys are hit, the more speed the runner gains. The only other key that comes into play is the space bar or "jump key" as it is



Harrier Strike Mission

PLAYABILITY

CHALLENGE

ADDICTIVENESS

EASE OF LEARNING

GRAPHICS

System: Atari ST
Price: $49.95
Summary: High-flying aircraft simulation/battle
Manufacturer:
Miles Computing
7741 Alabama Ave.
Ste. 2
Canoga Park, CA 91304
(818) 341-1411

The unique abilities of the Harrier Jump Jet make it one of the most innovative and versatile aircraft in history. It can be piloted like a jet at velocities near the speed of sound or maneuvered like a helicopter by directing engine thrust downward, allowing for hovering and vertical takeoffs and landings.

Harrier Strike Mission puts the controls of a Jump Jet in your hands and challenges you to destroy an enemy base and its tanks, aircraft, and fuel depot. Your arsenal includes three Sidewinder heat-seeking missiles, unlimited cannon fire, and 16 flares, which, by means of strong infrared output, draw enemy missiles to themselves and away from your aircraft.

This may not seem like much of a challenge, but remember, you must control the aircraft as well as the armaments. *Harrier Strike Mission* is an adequate flight simulator that will take some time to master. Sensitive mouse control and a fair number of keyboard commands will have you careening wildly about in the sky on your first few flights.

Fortunately, the creators have provided a quality manual and some options that will enable you to ease into familiarity with the controls: you can, for example, choose peacetime, unlimited fuel supply, or slower enemy speed to make your mission more manageable.

While the options include some graphic enhancements, one discouraging thing about *Harrier Strike Mission* is that there is only one mission to accomplish. Once you can finish the job on the Expert level, there is not much you can do to vary the challenge. Hopefully, Miles Computing will support buyers of the program with additional missions on disk, and a sequel is in the works. I'll be hovering patiently near my ST awaiting new challenges.

Note: At the time of this writing, Miles Computing was adapting *Harrier Strike Mission* to include joystick control. The new version was not ready in time for review, but should be available by the time you read this.—***Andy Eddy***

called in the manual.

What the game boils down to is banging as fast as possible on the keys and then picking the right time to hit the space . . . er . . . jump key. That's all there is to it.

Don't take that to mean the game is easy. Getting the runner up to speed takes great finger-to-key coordination and an ability to forget how much you paid for your computer. And hitting the jump key has to be timed just right. Should you fail on either of these counts, a little man with a clipboard will come by, look at your result and say something insulting. Of course, you don't actually hear him, but you can read what he says in a comic-strip balloon above his head.

Points awarded to each competitor are based on standards. Players compete for the best time, height, or distance in each event; then point scores from all events are combined to determine an overall winner. It is quite possible for a player to win seven of the ten events and not be overall champ if his scores were very poor in the other three events.

Practice is the key to success in this game. There isn't anything novel—or even very exciting—about *Arena*, other than the tremendous graphics, which will almost make you think you can smell the guy running next to you. But if you are into graphics and digital aerobics, you might want to give it a try.

—***Rick Teverbaugh***

Arena

System: Atari ST
Price: $34.95
Summary: Track and field simulation with outstanding graphics
Manufacturer:
Psygnosis Ltd.
Liverpool, England
Distributors:
Computer Software Service
129 Sherman St.
Cambridge, MA 02140
(617) 876-2505

495-A Busse Rd.
Elk Grove Village, IL 60007
(312) 439-4444

PLAYABILITY
CHALLENGE
ADDICTIVENESS
EASE OF LEARNING
GRAPHICS

Gato

Gato is a one-player simulation that gives you full control of and responsibility for a World War II combat submarine. In the game, you are given one of a series of missions that you must complete before you can return to the safety of Allied territory. Twenty missions are included on the disk, and Spectrum Holobyte promises that additional missions will be made available to extend the challenge of the game.

As captain, you must control the elements that keep your craft in good condition and your crew in good health—oxygen, battery, and fuel levels; speed and direction; torpedo and mine quantities, etc. Managing the sub is not an easy task, but you do have maps on hand to guide you—at least on the beginner levels. On higher levels—there are ten in all—it takes more shots to down an enemy ship, the enemy craft show up differently on your display maps, and the instructions for your mission are transmitted in Morse Code rather than being printed on the screen as they are when you start out.

Also variable throughout the contest are factors that affect both enemy ships (aggressiveness and speed) and your own sub (turning rate and hull strength). When you know what these factors are, you can tailor the battle to your own abilities and experience.

The choice of mouse or keyboard control and GEM compatibility make the *Gato* scenario very easy to set up and play.

Though the layout is easy to understand and the game is fun to play, there are a few shortcomings that may hinder your enjoyment of the game somewhat. Seeing a ship through your periscope when it is actually on the other side of an island, for example, is annoying, but you can make allowances and still be challenged by the game. Nor are sound effects and graphics used to anything resembling the full potential of the ST.

All in all, *Gato* is an entertaining underwater simulation, and I look forward to more releases from Spectrum Holobyte.—***Andy Eddy***

PLAYABILITY
CHALLENGE
ADDICTIVENESS
EASE OF LEARNING
GRAPHICS

System: Atari ST
Price: $39.95
Summary: Good submarine simulation that could use a little polishing
Manufacturer:
Spectrum Holobyte
2061 Challenger Dr.
Alameda, CA 94501
(415) 522-3584

Micro League Baseball

PLAYABILITY

CHALLENGE

ADDICTIVENESS

EASE OF LEARNING

GRAPHICS

System: Atari ST (and 8-bit)
Price: $59.95
Summary: Baseball simulation with arcade-like graphics

Manufacturer: Micro League Sports Assoc.
2201 Drummond Plaza
Newark, DE 19711
(302) 368-9990
(800) PLAYBAL

Since its introduction three years ago, *Micro League Baseball* for 8-bit computers has been flattered by many imitators. None has approached it in graphics and only a few have surpassed its statistical base.

Now, with the development of computers like the ST with larger memories and superior graphics, baseball fans have been clamoring for a new version of *Micro League* that would again take the lead in baseball simulations.

The new ST version answers that call but not always with all the features for which the fans have been longing.

The ST version is clearly superior to all other versions in four areas. The first is in the area of unusual plays. Making use of the extended memory available, almost anything possible in a real game can now occur in *Micro League*. These plays vary from sublime interruptions for minor injuries to more radical changes for ejections, rain delays, and power failures (mostly in domed stadiums).

A second improvement comes in the area of customizing each game to the stadium of the home team. The program takes into account not only the dimensions of the park but whether there is natural grass or artificial turf.

Based on a small sample of games, it appears that there are more base hits through the infield when a contest is played on turf rather than grass. You would also expect errors to be cut down due to truer hops.

A third area of improvement is also an area of concern. With the original *Micro League* it was necessary to preload another disk before the game disk if you wanted to print out a box score or compile stats for the game to be played. Now that decision isn't necessary because both functions are included on the game disk—a real plus.

But in the original version it was also necessary to buy a General Manager's

Match Point

PLAYABILITY

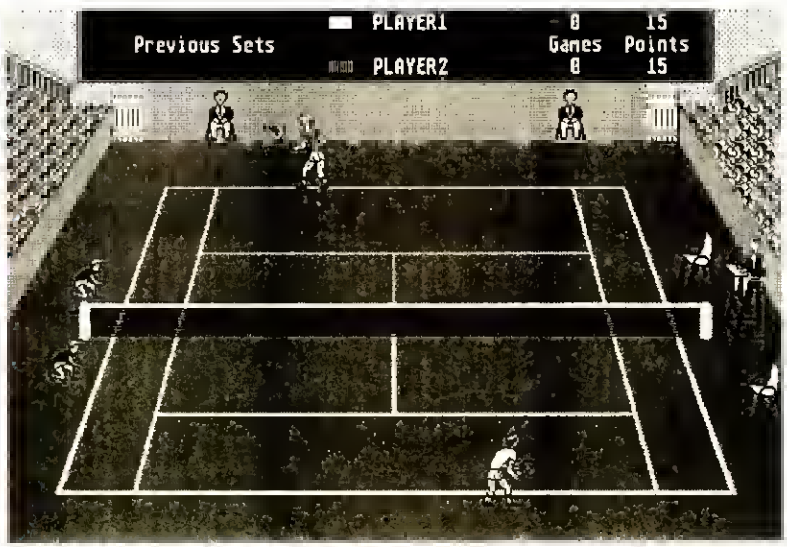
CHALLENGE

ADDICTIVENESS

EASE OF LEARNING

GRAPHICS

System: Atari ST
Price: $39.95
Summary: Arcade-type tennis simulation
Manufacturer:
MichTron Inc.
576 S. Telegraph
Pontiac, MI 48053
(313) 334-5700



Tennis isn't a complex game: two people with rackets hit one ball back and forth over a net. Should be a snap for a computer to simulate, right? Wrong, evidently. Even this current effort, *Match Point*, leaves the sport of tennis without a solid simulation on the computer market.

The Nintendo people have a good one in the arcades and for their own home game system, but computers still haven't found a champion.

Match Point is a near miss in several areas and a blatant double fault in others. As you might expect in an ST product from MichTron, the graphics are nearly flawless. It is quite easy to follow the flight of the ball thanks to the well-drawn figures and the realistic shadow.

There are ballpeople to retrieve the ball when it goes into the net and line judges who are so accurate that even John McEnroe would probably find little reason to complain.

After each odd-numbered game, the players trade sides. The match is a best of five sets, so hang in there. No lead is

disk to create teams of your own, and then you were limited in the number of teams that could be saved at one time. Expanding beyond that number meant buying another Security disk.

Fans had hoped that the new version of *Micro League* would recognize the shortsightedness of that policy and make it possible to save teams on a blank formatted disk and have the creation program as part of the main package. Alas, nobody at Micro League was listening to the roar of the crowd.

The final area of improvement is in graphics. There wasn't anything wrong with the original format, but the ST version is much improved. The sliding of the runners is better drawn, and the fielders really move their arms to throw the ball.

There is one more change that can't really be considered an improvement. The program will accept commands from the mouse. It isn't easy to understand why anyone would choose to press the mouse button, move the pointer to the proper spot, and press the button again instead of merely selecting one key on the keyboard, but for those who have forgotten how to use the keys, the feature is there.

For those not familiar with the original version, here is a quick recap of the features that have made *Micro League Baseball* an 8-bit institution. Twenty-five teams, ranging from Hall of Famers to a couple of all-time great franchises, are included on the disk as are some great individual seasons, beginning with the 1927 Yankees and extending through the 1983 Orioles.

When his team is at bat, the manager can choose to swing away, bunt for a hit, run aggressively or safely, steal, hit and run, sacrifice, send up a pinch hitter, or send in a pinch runner. On defense the options range from type of pitch (fastball, curve, slider, change) to pitchout, intentional walk, moving the infield in at the corners or in all around, and making lineup changes. You can also have your pitchers warm up in the bullpen; the new version tells you when the pitcher is ready or when he has toiled too long in the pen.

One other addition, good for league play, is the option to have the computer handle both sides of a matchup, taking away the graphic display and removing all pauses. In this way, you can play a game necessary for the standings and stats of the replay but not of play-by-play interest to the gamer in under five minutes.

All things considered, *Micro League Baseball* is a worthy addition to the library of any ST owner who loves baseball.—***Rich Teverbaugh***

insurmountable, unless you're playing against the computer.

And that brings us to one of the faults. There are three levels of difficulty in taking on the computer, but none seems tame enough to really introduce the game. Not a novice at sports games by any standard, I played against the easiest level and wound up with a score of 6-0, 6-1, 6-0 in my first match.

The selection screen appears after the game boots. The computer can play both sides (nice for a demonstration but after that, who cares?), one side, or none at all. Control can be by keyboard (an awful choice), mouse, or joystick. Because most games of this type are played with the joystick, this method will be the most familiar in the shortest time, but I recommend the mouse, especially if you have a good pad for the critter.

Really, the biggest problem with *Match Point* is the rule book, the area easiest to correct yet most often bungled by software houses. The actual rules of game play take up two pages. There is no discussion of how to control the direction the ball takes as it leaves the racket, nor are there any instructions on how to serve—what hitting the button at the height of the toss means as opposed to waiting longer, for example.

There is a pause key that works all the time and another key to end the game that works only on occasion. Four versions of the game are found on the disk—monochrome and color versions in both English and French. This provides a popular solution to the old homework vs. play dilemma—practice a foreign language while you improve your eye/hand coordination.

I would have preferred it had MichTron used the disk space to make a doubles game available. It probably wouldn't be possible to make it a four-player all human contest, but the computer could have handled the teammate of each human foe or two teammates against two human foes.

In all, the game does simulate tennis action. It is easier to rally from the baseline than take chances and charge the net, for example, but it is nearly impossible to beat the computer with that conservative strategy. *Match Point* is a pleasant diversion, but definitely not a Grand Slam event.—***Rick Teverbaugh***

Strategy Games Old And New

By ARNIE KATZ and BILL KUNKEL

Mind challengers from Chess to Archon for Atari 8-bit computers

The Atari 8-bit computer line is more than just an affordable home arcade machine. Its excellent audio-visual capabilities are tailor-made for thunderous explosions, sizzling lasers, and adorable animated characters.

But the library of Atari entertainment software includes a lot more than just slam-bang shoot-'em-ups and thrill-a-second maze chases. The catalogue bulges with programs designed to test your brain rather than your motor reflexes. In fact, there are so many pure strategy games that even a lengthy article could only scratch the surface. If we are to do justice here to the best of the brain games, we must leave the wargames, sports simulations, and adventures to future issues of *Atari Explorer*.

Similarly, programs that require a lot of manual dexterity are beyond the scope of this survey. Action-strategy contests like *Rainbow Walker* from Epyx and *Boulder Dash* from Avantage will get their due in an upcoming article about action-strategy games for the 8-bit systems.

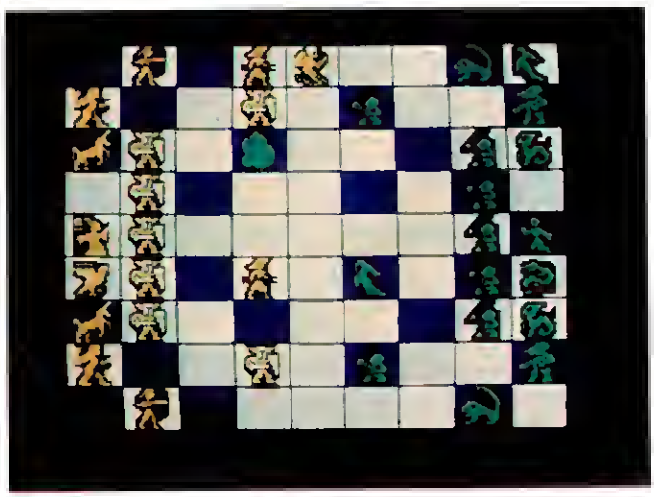
The King of Strategy Games

After hundreds of years and millions of minutely annotated matches, chess is indisputably the most popular and respected strategy game of all time. Its essentially simple rules and breathtaking range of possible moves keep it fresh and fascinating through a lifetime of play.

When "electronic brains" made the transition from science fiction to daily newspaper headlines after World War II, it was only natural to apply the logic-handling capabilities of computers to chess, the most intellectual of all board games. Computer scientists found the idea attractive, too, because a chess-playing machine could provide the average American with a comprehensible, yet impressive demonstration of the power of microprocessing.

Computer chess has come of age in the last decade, and numerous tournaments pit computer chess players against each other or, occasionally, human opponents. Though today's chess programs play much better than those published even as recently as the early

Chessmaster 2000



Archon

1980s, a program that can defeat a Master or Grand Master is still only a dream.

Computer chess has several advantages over the non-electronic version of the game. It provides an ever-present source of opponents, it offers a context for learning and improving your proficiency, and it can be fine-tuned to give any chess lover a competitive game.

Sargon III from Spinnaker Software, designed by Dan and Kathe Spracklen, is the most recent version of the classic home computer chess program. Whether you are an expert or a novice, *Sargon III* is ideal for times when you want a game and can't find a human foe. Its nine difficulty levels expand to 18 choices with a special setting that limits thinking time for the computer. You can even raise or lower the skill level during a game.

Sargon III can moderate a match between you and a human rival, compete with you, or even play against itself. The latter mode allows novice players to study moves and situations without the strain of active participation.

This high quality chess disk provides hints and lets you take back moves, switch sides, invert the board, or alter the positions of the pieces. You view the board from an overhead perspective, with pieces shown in profile.

Sargon III also contains a roster of famous games and problems for serious chess enthusiasts to replay or study. Other programs now claim to be able to beat *Sargon III* at the top skill setting, but few offer as much for the average-to-good chessplayer.

The top-ranked challenger to *Sargon III* is *Chessmaster 2000* from Software Toolworks. The package boasts of tournament victories over rival programs, but what really distinguishes *Chessmaster* is its flexibility and wealth of features. Simply put, no program offers a broader range of difficulty settings or helpful features.

Chess Nouveau

Chess may be the top strategy board game of all time, but that doesn't keep people from trying to improve it. Over the years, game authors have proposed hundreds of variations, called Fairy Chess, on the basic game, and a few have tried to apply the overall principles of chess to new environments and situations.

The most intriguing attempt to create a "new chess" for the Atari is *Archon* from Electronic Arts. It is the most significant computer contribution to the field of strategy-oriented board games.

Archon, which may have been inspired by the holographic board game C-3PO and Chewbacca played in "Star Wars," is a battle between the armies of Light and Darkness. The armies, though of equal power, are not composed of identical pieces. Icons include a unicorn, phoenix, valkyric, golem, djinni, and wizard for the Light forces, while the banshee, troll, basilisk, shapeshifter, dragon, and sorceress represent the Dark side. There are also several kinds of pawns on each side.

Only the existence of the computer makes the *Archon* board practical. Some squares on the board cycle from Light to Dark. The lighter a square, the more it enhances the power of any Light piece sitting on it. Dark pieces benefit most from similarly colored squares. These shifts keep the game situation fluid and put a great premium on planning ahead to take advantage of upcoming color changes in key positions.

Unlike chess, *Archon* uses an action-combat sequence to determine the results of combat. When two icons occupy the same square, the combatants are transported to an arena screen where their particular strengths and weaknesses combine with your joystick aptitude to settle the issue.

The graphics are excellent. Each of the individualized icons moves and fights in a unique way, and they are easily distinguished from each other on the board and in the arena.

While *Archon* requires a little expertise with the joystick to win the battles, it is primarily a strategy contest, and a challenging one at that. It provides a refreshing change of pace for chess fans.

Free Fall Associates, the design team behind *Archon*, has produced a sequel, *Archon II: The Adept*, also from Electronic Arts. It makes a noble attempt to range further from chess than its predecessor, but it is not quite as satisfying to play.

Archon II is a battle between the Master of Law and the Mistress of Chaos. Each side employs four magicians, called adepts, to summon demons and elementals, which can attack the rival force. The adepts can also cast spells which heal, imprison, weaken, release, or banish creatures which rival adepts have conjured. There is also an "apocalypse" incantation, which forces a lopsided match to a speedy conclusion.

The most prominent features of the

Archon II


Lords of Conquest

After hundreds of years and

millions of minutely annotated matches, chess is indisputably the most popular and respected strategy game of all time.

main game board are the four bands that represents the elements of fire, air, water, and earth. In addition, there are two neutral "void" squares and a citadel for each side. Adepts can teleport anywhere on the board, though at some cost in magical energy, while icons slide along the element band as far as desired to jump to a different band.

As in *Archon*, combat is settled on a separate screen. The powers of the various demons and elementals come into play as players move and fight using a joystick-activated control scheme.

Archon II: The Adept offers a unique strategy gaming experience. Its game board, so different from the rows and files of squares of a traditional chess board, produces a freewheeling contest in which tremendously powerful pieces can move around the board quickly and strike almost anywhere. It is, admittedly, the second squeezing of the grapes, but *Archon II* is still a heady vintage.

Strategy Conquers All

The territorial imperative is the guiding priciple behind a pair of geopolitical strategy games—*Lords of Conquest* from Electronic Arts and *Colonial Conquest* from Strategic Simulations.

Both feature more realistic settings than the abstraction of the chessboard, but the attraction for strategists is quite similar in many ways. Like chess, these games take a broad view of the situation, rather than simulating a wealth of small details as wargames do.

Eon Software, which previously designed such non-electronic strategy games as Cosmic Encounter, has made the most of the switch to the computer. In *Lords of Conquest*, one to four participants struggle for no less a prize than world domination, and, in our opinion, the game surpasses all similar non-electronic board games.

One of the outstanding features of the game is the module that lets you create additional mapboards when you get tired of the 20 that come with the package. So when you become jaded by one playfield, there is always another to offer fresh strategic possibilities.

Lords of Conquest can be played at any of four levels of complexity. While only two types of resources, gold and horses, are important at the easiest setting, the more complex versions add additional resources, allow a country to shift resources from territory to territory, and permit ship construction.

Players represent imperialistic powers. During each turn, equivalent to a year in real time, countries try to seize territories to acquire their resources. Raw materials allow a nation to build cities, and the first player to construct three and defend them against rivals for a year wins the game.

Although the program can provide automated opposition, Lords of Conquest is best when played with human opponents. Diplomatic dealing is the most fascinating aspect of the game, and that requires the presence of other people.

Colonial Conquest from SSI is a multi-player strategy game based on the struggle for world domination in the period 1880-1914. One-to-six players assume leadership of six major countries and try to gobble up more than 120 minor nations. England, Germany, France, the United States, Japan, and Russia must use diplomacy, covert intelligence, and military might to conquer territories and gain victory points.

One by one, players take control of the screen and use the joystick to enter orders for the coming quarter. They can build armies, fleets, and fortifications; dispense foreign aid; engage in espionage; pay bribes to de-stabilize minor countries; move land and sea forces; and engage in combat.

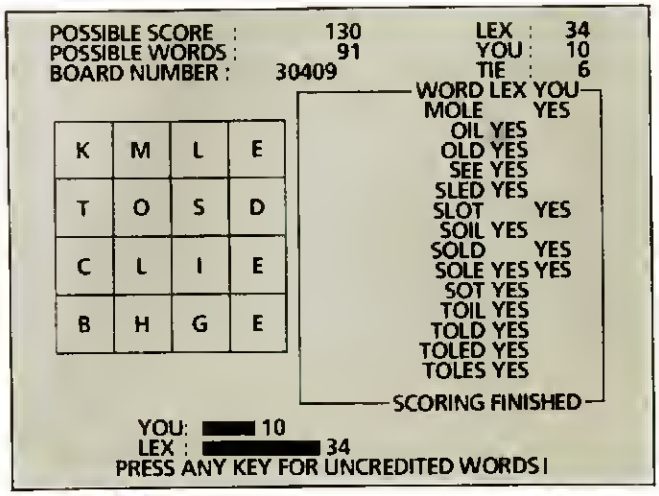
The disk includes a choice of three initial set-ups. The standard scenario starts each major country with no holdings beyond its national borders. The 1880 and 1914 versions of *Colonial Conquest* are historical. They assign each major country the possessions it actually controlled in the indicated year.

If you like non-electronic strategy contests like Risk and Diplomacy, be sure to check out these titles. They have a similar feel, augmented by the extra features that can be found only in computer games.

A War of Words

The package for *Word Challenge* from Spinnaker bears the inflammatory phrase, "Better then Boggle." The game inside not only equals Parker

Colonial Conquest



Word Challenge

Brothers' outstanding tabletop game, but employs the computer to surpass it in several respects.

The object is to find and list words of three or more letters which are hidden within a grid of letters. Each letter of a word must be adjacent to the next letter in the word horizontally, vertically, or diagonally on the letter matrix.

A robot adversary, Lex, can be handicapped to make the game a reasonable challenge for any player. At game's end, the program counts up the points and tallies the scores.

The optional features make a great word strategy game even better. *Word Challenge* has 26 difficulty levels and modes which allow everything from credit for "embedded words" (*run* is "embedded" in *runner*) to manual setup of the board. The size of the playfield itself can be 3 × 3, 4 × 4, or 5 × 5.

The often-used phrase "fun for the whole family" describes *Word Challenge* perfectly. Either solitaire or against live opposition, this program will please anyone who enjoys word play.

The Lighter Side of Strategy

Even avid strategy gamers need a change of pace, and *Strip Poker* from Artworx may fill the bill. This simple-to-play poker game is considerably enlivened by the nature of the computer controlled opposition.

When the player, shown in full-color illustration, runs out of chips, he or she trades in an article of clothing for more chips. If you show good "card sense," it doesn't take long before the bare facts become obvious.

A selection of opponents is available. A couple are included with the basic game, and additional data disks can provide variety. Although most of the on-screen strippers are attractive ladies, Artworx also markets a male poker partner.

The Atari 8-bit computers will always have a special place in the hearts of joystick jockeys everywhere, because they present such vivid action games. But as we have seen, more thoughtful Atarians should also esteem the 400/800/XL/XE range for its mind-stretching strategy software. ■

Where to Find the Games

Artworx
1844 Penfield Rd.
Penfield, NY 14526
(800) 828-6573

Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
(415) 571-7171

Software Toolworks
13557 Ventura Blvd.
Sherman Oaks, CA 91423
(818) 907-6789

Spinnaker Software
One Kendall Square
Cambridge, MA 02139
(617) 494-1200

Strategic Simulations
1046 N. Rengstorff Ave.
Mountain View, CA 94043
(415) 964-1353

Data Transfer:

The Productivity Bridge

Need to transfer data between your Atari and another brand of computer? Here are some strategies that can help.

You have just purchased a shiny new Atari ST and are planning reluctantly to retire your true-blue, much-used Atari 8-bit system. Unfortunately, half your novel is already stored on 5.25″ disks in *Atariwriter* format.

Or perhaps you have an IBM PC at the office and an ST at home, and you want to take work home on a regular basis.

Or maybe you have found a BBS simply loaded with public domain clip art for the Amiga—just the kind of stuff that could make your latest *NeoChrome* graphics presentation a big hit. What do you do?

There is a big computing world filled with useful information out there. Unfortunately, given the number of different computer and software brands and the resulting welter of disk and data format standards, a good part of this information isn't immediately available to your Atari.

For some, this is a hindrance to productivity and a source of frustration. But it doesn't have to be. Many methods that give Atari users access to data—and in some cases, even software—used by "incompatible" machines exist today. This two-part series describes strategies that can help you and your Atari reach out into the wider world of information at large.

Making the Connection

The first phase of the data transfer process involves moving data from one incompatible system to another. This can be done in several ways.

If you want to move data between your Atari and a system such as an IBM PC or Macintosh (particularly on a frequent or high-volume basis) the best all-around solution may be to equip the machines to read one another's disks directly. Data transfer then becomes a matter of carrying disks back and forth between the two computers—ideal if, for example, one system is at the office and the other at home.

Surprisingly, an ST equipped with double-sided drives is quite capable of reading and writing to IBM PC format 3.5″ disks. IBM-to-ST transfer problems can thus be solved by equipping your PC or close compatible with an IBM-style 3.5″ outboard disk drive and controller card (or using an IBM Convertible PC). One caveat: PCs some-

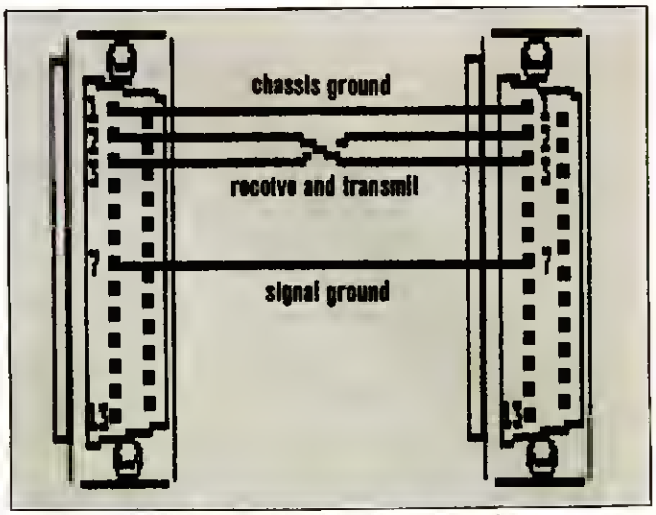
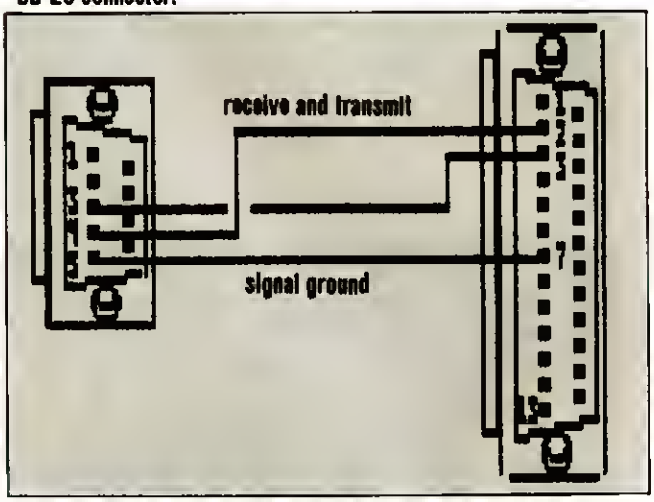


Figure 1. Schematic of a "standard" null modem cable with 25-pin connector at each end.

Figure 2. Schematic of a null modem cable—Atari 850 interface to DB-25 connector.



Parts list (check your system to determine proper gender for connectors):

- **Male DB-25 connector (Radio Shack #276-1547)**
- **Female DB-25 connector (Radio Shack #276-1548)**
- **Male DB-9 connector (Radio Shack #276-1537)**

By JOHN JAINSCHIGG

times have trouble reading disks formatted on the ST, so use disks formatted by the IBM for all transactions between the two machines.

An alternative approach is to buy an IBM-compatible 5.25″ floppy drive for your ST. The Microbyte A and B from Paradox Enterprises and the I.B. Drive from I.B. Computers, both let STs read, write, and format IBM XT and AT-standard disks. The Atari PC, available this summer, will offer the best solution in both directions—it comes with a 5.25″ drive built in, but can work directly with ST 3.5″ drives as well. File transfer heaven!

The Macintosh connection is currently a bit more problematic. Pacific Data Systems is said to be developing a device that will give an ST the ability to read and write to Macintosh disks. Data read can then be saved on ST-format disks for processing by the ST in its native mode. No release date has yet been set.

Atari 8-bit systems are not as well supported as STs in the emulation department. A device called the ATR8000, no longer manufactured, allowed an Atari 8-bit computer to interface with standard 5.25″ and 8″ drives, enabling disk exchange with CP/M-based systems. ATRs may occasionally be found through mail-order or second-hand hardware concerns.

A more flexible approach to data transfer involves getting computers to talk to one another by wire. The same kind of software and equipment that lets your Atari communicate with bulletin board systems and information services can serve to transport data between it and another, similarly equipped small computer.

Direct Connection

If the systems you are looking to connect are close together and both have RS-232 ports, they can be linked via a simple *null modem* cable.

(NB: while Atari 8-bit systems do not have RS-232 ports built in, these can be added inexpensively by purchasing an Atari 850 or ICD PR:Connection general-purpose interface unit. The alternative, attempting direct connection through some medium other than standard RS-232, requires technical expertise greater than that possessed by most casual users and is not well-supported by commercial products.)

A null modem cable can be put together by anyone with a flair for soldering or made up by a cable shop according to the schematics in Figures 1 and 2. Figure 1 describes a "standard" null modem cable with a 25-pin connector at either end, while Figure 2 shows a variation for 8-bit owners who use the Atari 850 interface module.

Note that the only difference between a null modem cable and a "straight through" RS-232 cable of the kind used to hook up a modem is that connections to pins 2 and 3 (transmit and receive data) are switched at one end. This serves to conduct data transmitted from one computer to the "receive data" pin of the other, and vice-versa.

Once the machines are connected in this fashion, standard terminal software is used to send files back and forth between them. Terminal parameters must be the same for both systems—a good "generic" setting is 8 data bits, 1 stop bit, no parity, half duplex. Baud rates must also be identical and can be set to the highest speed both systems can manage—usually 9600 baud.

Because the systems are close together, you can control both ends of a file transfer yourself with little difficulty. For additional convenience, one system can be designated as *slave* and its terminal software replaced with a bulletin-board package, permitting remote control.

If both systems are already equipped with Hayes-compatible modems, an alternative method of direct-connect can be implemented by running a modular telephone cable between their line jacks. Hayes-compatible modems are unusual in being able to perform this trick—most modems cannot. Note that in this configuration, data transfer speed is reduced to whatever maximum speed the modems can support.

Remote Connection . . . Alternatives

If the machines you want to connect are in different locations, communication by modem and phone line may be the only way to go. While the prototypical configuration for this kind of telecommunication involves modems, terminal software, and trained operators at both ends, this may not be the most efficient approach in all situations. Among other problems, it involves the logistical complication of getting someone else in on the act—not always an easy thing to do.

If a pair of modems is already available and at least one of them has auto-answer capability, one way to eliminate the second operator is to run a BBS package on one machine. While convenient, however, this approach will probably force you to leave the slave system up and running for long periods of time—not great for the equipment. Moreover, while the BBS is up and running, your data will be vulnerable to scrutiny by, shall we say, "unauthorized" callers. Using a BBS package that provides password protection can limit this risk considerably.

Another way to eliminate the second operator is to adopt a sequential mode of transfer in place of "real-time" telecommunications. To use this approach, you must have access to the upload/download sections of a public BBS or, better yet, have an account with a commercial information service like Compuserve or Genie.

Step 1 is to upload your files from the source system to the BBS or service, using, if possible, a *personal file area* to which others who log on will not have access. Later, you can download your files using the target system. Note that if both source and target systems can use the same modem, only one modem is needed to pull off this kind of transfer—just take it with you when you go.

In a worst case situation ***some serious translation may have to take place before data can be used.***

A portable computer like the Radio Shack Model 100 can also be used to ferry data between source and target systems, eliminating the need for modems and additional personnel. In this scenario, data is transmitted from the source system to the portable using a direct-connect cable and standard terminal software, as described above. Then, the portable is carried to the target system and the process repeated in reverse.

While this may seem like overkill, you would pay considerably more for a pair of Hayes modems than you would for a Model 100 . . . not that I'm trying to rationalize, but those portables sure are fun to own!

Data Types

Now that we have examined our options for getting data from one machine

to another, let's begin looking at phase two of the transfer process—presenting the data in acceptable form to software running on the target system.

This almost always entails some kind of data massage. In a best case situation, mere *filtration* may be all that is required. For example, if an Atari 8-bit system is sent a standard ASCII text file, care must be taken to convert all ASCII carriage returns (decimal 13's) to 155's as per Atari ASCII (ATASCII) requirements.

In a worst case situation, by contrast, some serious translation may have to take place before data can be used. For example, a picture file created on an Atari 8-bit system using *MicroIllustrator* software must be subjected to the following manipulations before it can be used on the ST low-res screen under *NeoChrome*:

1. Color register and other information must be separated from the actual picture file.

2. The picture must be "uncompressed" to its original, hardware-dependent, graphic bitmap form.

3. The bitmap must be scaled to expand it from Antic E mode resolution to Atari low resolution.

4. Color equivalencies must be calculated and a *NeoChrome* compatible RGB table set up.

5. The scaled bitmap must be remapped using four bits per pixel instead of two in such a way as to conform with the ST low-res screen memory layout.

6. A *NeoChrome*-compatible picture file, containing color information and the revised bitmap, must be assembled.

... Agony, or what? No wonder "graphics standards" are in the news nowadays.

The amount and type of processing required before a transferred file can be used is proportional to how closely the fundamental data type (text, graphics, sound, etc.) is tied to hardware features. To understand what this implies, let's look at some examples.

Most computers provide operating system facilities for handling text in ASCII form—ASCII text is thus a generic data type, not bound by a particular hardware system. As a result, programs that generate text (editors, word processors, etc.) tend to be able to share data with a fair degree of freedom, regardless of the system they are running on. Moreover, even programs like spreadsheets, because they ultimately produce what can be thought of as "readable material," can share data in text form, though their "native" file formats may differ alarmingly.

Graphics data, by contrast, is very closely aligned with specific aspects of the computer hardware. Total incompatibility is the rule, even between different graphics modes of the same system. Moreover, because graphics usually occupy a lot of room in memory and on disk, graphics programs tend to implement baroque schemes for file compression that further complicate the lives of data transfer mavens. Sound data is similarly hardware-dependent.

Next issue, we'll put together the complete data-interchange toolkit—a group of commercial, public-domain, and, yes, even home-made utilities that will perform Shiatsu massage on even the most intractable information. ■

Part 4: An Intro to BIOS I/O

Programming with GEM

By JOHN JAINSCHIGG

In preparation for next issue's planned foray into the world of high-level GEM user interface programming—events, dialog boxes, windows, etc.—we thought it would be a good idea to pause and explain a few of the lower-level I/O functions of the ST, particularly those that have cropped up without explanation in previous example programs in this series. Most of these have been functions supported by the ST BIOS (Basic Input/Output System).

As before, our medium of exploration will be the C programming language—specifically, the Mark Williams implementation—but users of Pascal, Modula-2, and other languages that support a consistent interface to low-level OS functions will have little difficulty abstracting from this discussion to their own preferred programming environments.

So far, each article in this series has concluded with one or more large-scale example programs. This time, to add some variety and change the pace, we have scattered short demonstration programs, utilities, and functions throughout.

BIOS I/O Functions

BIOS functions available through Mark Williams C include macros for handling single-character I/O with the console (keyboard/screen combination), printer, RS-232 (auxiliary), and MIDI ports.

BIOS macros also exist for examining the state of the Shift, Alternate, and Control keys; determining the logical system configuration; inquiring about the media change status on a particular drive; reading and writing individual sectors on a disk; and setting 68000 exception vectors for low-level error trapping and other purposes.

BIOS-level I/O, while crude by comparison with the far more elegant, all-encompassing functions provided by higher levels of the OS, is fast and easy to use. Often, a handy BIOS call is just the thing when you're trying to hack something up in a hurry. For example, Listing 1 shows the program I use to set my SMM804 printer to boldface mode, prior to printing out text files from the desktop.

That's all there is to it—three BIOS-level calls. No need to worry about opening workstations or initializing AES control arrays—just good old, down and dirty I/O. The program can be compiled from the $ prompt of the Mark Williams MSH shell by typing:

```
$ cc -o boldprt.tos boldprt.c
```

where boldprt.c is the name of your source file. The resulting object file, boldprt.tos, can be executed from the shell or by double-clicking on the filename.tos icon from the desktop.

In Listing 2 is another example of low-level BIOS I/O programming: a (very) dumb terminal program that will let you communicate with Compuserve and other full-duplex information services.

Compile the program by typing:

```
$ cc -o dumbtrm.tos dumbtrm.c
```

where dumbtrm.c is the name of your source file. The resulting object file, dumbtrm.tos, can be executed from the shell or the desktop, as you prefer.

In compiling both of these programs, we have used the -o option of the shell to put a .TOS extension on the output files, instead of the default .PRG. Programs with .TOS extensions are launched somewhat differently than programs with .PRG extensions.

.PRG programs are launched into the GEM environment: the screen is blanked, assuming the foreground color; the mouse cursor is switched to busy bee format; and the name of the program is written into the center of the menu bar. In contrast, by giving the object file a .TOS extension, we can cause it to be launched in a more subdued manner, into a lower-level TOS environment.

When a .TOS application starts up, the screen blanks out, changing to the background color; a square block cursor appears in the upper left hand corner; and the mouse cursor disappears.

The TOS environment emphasizes simple character display and keyboard control, rather than graphic interactions. In other words, .TOS programs use the ST as if it were a (gasp!) conventional computer, which is fine and dandy for quick, low-level hacks and certain kinds of specialized applications that don't use the features of GEM.

Basic BIOS I/O Macros

The little programs in Listings 1 and 2 make use of four BIOS I/O macros—Bconstat( ), Bcostat( ), Bconin( ), and Bconout( )—defined in the OS bindings header file, osbind.h. Respectively, these macros permit input and output status testing, and single character input and output for the printer, serial port, console, and MIDI ports.

In using these macros, the different devices are referenced numerically, according to the following scheme:

0. Printer
1. Auxiliary port (RS-232)
2. Console (keyboard and screen)
3. MIDI ports

Bconstat( ) is used to tell if a character is waiting for input from a particular device. The device number is passed as an argument, and Bconstat( ) returns a non-zero value if a character is available; zero otherwise. The expression Bcon-

Listing 1.

```
#include <osbind.h>

main()
{
     if (Bcostat(0)) {    /* If printer is ready ...      */
          Bconout(0,27);  /* send it an ESCape ...        */
          Bconout(0,69);  /* and an 'E' to set boldface. */
          }
     exit(0);             /* Clean, non-GEM exit. */
}
```

```
#include <osbind.h>

main()
{
     char c;

     /* As long as 1 is true, i.e. forever ... */
     while (1) {
     /* If there's a character waiting for input
        at the RS-232 port ... */
     if (Bconstat(1))
     /* Retrieve it, mask off the top bit, and
        print it on the screen. */
          Bconout(2,(c = Bconin(1)) & 127);
     /* If there's a character waiting for input
        at the keyboard ... */
     if (Bconstat(2)) {
     /* Retrieve it. If the user has pressed ESC
        (decimal 27), then end ... */
          if ((c = Bconin(2)) == 27) break;
     /* Else send the character out the RS-232 port ... */
          Bconout(1,c);
     /* Remove the comment delimiters around the
        next statement if you want the program
        to work in a half-duplex environment. */
     /*   Bconout(2,c)                    */
          }
     }
     exit(0);
}
```

Listing 2.

stat(0), which would check for input from the printer, is naturally illegal. To check whether a character is ready for input at the RS-232 port, for example, our dumb terminal program uses the expression

```
if (Bconstat(1)) { ...
```

Bcostat( ) does the reverse—it checks to see if a device is ready to accept a character of output. Again, the device number is passed as an argument, and a non-zero value is returned if the device is ready; zero otherwise. Our printer-setter uses the following expression to check if the printer is ready to receive output:

```
if (Bcostat(0)) { ...
```

Bconin( ) handles actual input. The macro accepts a device number as an argument (Bconin(0), for the printer, is illegal) and returns a long (four-byte) value. When input is retrieved from the console (device 2), the low byte of the high word of the returned value contains the scan code for the key pressed, while either the ASCII code of the key, or zero, is returned in the low byte of the low word, depending on whether the key was ASCII or one the special ST keys. When input is retrieved from other devices, the byte of data is returned in the low byte of the low word, and the rest of the long can be discarded.

Bconout( ) performs output. The macro accepts two arguments, a device number, and an integer value for the character you want sent to the device. To send an ESC character to the printer, for example, our printer-setter uses the expression

```
Bconout(0,27);
```

where 27 is the ASCII code for ESC.

BIOS I/O Tips

As noted above, Bconstat( ), Bcostat( ), Bconin( ), and Bconout( ) are #defined as macros in the header file osbind.h. They actually represent four different kinds of call to a single real function: bios( ). This function accepts one of 13 available commands and one or two parameters as arguments, sets things up, and executes trap #13, dispatching one of the range of BIOS operations available. Bconin( ), for example, calls BIOS function 2. It is defined in osbind.h as follows:

```
#define Bconin(e)    bios(2,e)
```

If desired, the function bios( ) can be called directly, and examination of the macro definitions in osbind.h will reveal the command numbers and syntax required for each operation. If you decide to take this approach (out of machismo or love of the esoteric), you can drop osbind.h, provided you remember to declare the function bios( ) as returning an external long integer at the head of your source file:

```
extern long bios();
```

When using BIOS-level I/O, it is important to test input and output status for a device before attempting, respectively, to read from or write to it. If called before their time, Bconin( ) and Bconout( ) will wait, hanging up your program, until the device can perform the requested operation.

BIOS-level console I/O (device 2) is extremely handy for keyboard-oriented program control. Following are some handy functions that use console I/O to accept keyboard input in various ways:

▪This function waits for a keypress to occur, then returns the long value retrieved.

```
long wait_for_key()
{
     long c;

     while(Bconstat(2));      /* Clear the input buffer */
     while(!Bconstat(2));     /* wait for a fresh keypress ... */
     return(c = Bconin(2));   /* and return it to the caller. */
}
```

▪This function waits for a keypress to occur, returning an integer containing the ASCII code of the key pressed or zero if non-ASCII input was received.

```
int wait_for_ascii()
{
     int c;

     while(Bconstat(2));      /* Clear input buffer */
     while(!Bconstat(2));     /* wait for keypress */
     return(c = Bconin(2));   /* and return low word to caller */
}
```

Note that breaking out the low word of input returned is done by accepting input into a integer variable, effectively ignoring the high word of the returned value. This can also be done with a variable of type char.

▪This function can be used inline to grab a keypress on the fly. If no key has been pressed, the function returns a 0L (long zero).

```
long grab_key()
{
     long c = 0;

     if(Bconstat(2)) c = Bconin(2); /* If a key's been pressed */
     return(c);                     /* return it. */
}
```

▪This function, and its associated union and structure, demonstrate another way of breaking down BIOS-level console input into its various parts. When called, it waits for a function key to be pressed and returns an integer between 1 and 10, corresponding to it. Other keys are ignored.

```
union {
        struct {
                int SCAN;
                int ASCII;
                } key;
        long c;
        } keyin;

int get_function_key_number()
{
     do {
          keyin.c = Bconin(2);
          } while (keyin.key.SCAN < 59 || keyin.key.SCAN > 68);
     return(keyin.key.SCAN - 58);
}
```

BIOS-level Console Output

When using BIOS-level console output, the following expression:

```
Bconout(2,13);  /* 13 is ASCII carriage return */
```

will cause the cursor to return to the left margin of the screen, but will not move it down one line. To do this, an explicit linefeed must be sent, using

```
Bconout(2,10);  /* 10 is ASCII linefeed */
```

This is somewhat confusing for C programmers accustomed to using standard I/O functions such as printf( ) for screen output. When a newline (\n or ASCII 13) character is sent through printf( ), the result is a carriage return with linefeed. The standard library function adds this linefeed automatically.

The VT52 Terminal

One of the most interesting features of BIOS-level console output is that it provides an easy-to-use and elegant system of screen controls—controls to move the alpha cursor around, blank all or part of the screen, and change character foreground and background colors. These functions are actuated by sending special *escape sequences*—groups of one or two characters preceded by the ESC character, decimal 27—through Bconout(2). The available functions correspond to features of a VT52 terminal and are referred to collectively as "VT52 commands."

The short program in Listing 3, when run on a color monitor, demonstrates some of the VT52 commands.

Compile the program, double-click on the icon, and . . . instant greenscreen (assuming that you haven't changed the default colors of your desktop). The function con_string( ) sends a null-terminated string out through Bconout(2).

Depending on resolution, determined by Getrez( ), the program outputs one or another string of VT52 escape sequences. The first sequence sets the foreground color of characters to index 2 (normally green); the second sequence sets the background color to index 3 or index 15, depending on resolution (normally black); and the third sequence clears the screen (filling it with all-black space characters) and places the cursor in the upper left-hand corner (at the home position).

Note that the expression \33 is octal string notation for ESC (decimal 27) and that other numbers are expressed in octal as well. If you feel uncomfortable with this notation, a sequence of Bconout( ) commands, with decimal or hexadecimal arguments, can accomplish the same thing:

```
Bconout(2,27);     /* ESC */
Bconout(2,'b');    /* b -- sets foreground color */
Bconout(2,34);     /* Color index 2 + 32 */

Bconout(2,27);     /* ESC */
Bconout(2,'c');    /* c -- sets background color */
Bconout(2,35);     /* Color index 3 + 32 */

Bconout(2,27);     /* ESC */
Bconout(2,'E');    /* E -- clears screen, homes cursor */
```

A complete list of screen control sequences is given in the Atari developer's documentation and on page 446 of the Mark Williams v1.04 lexicon (under Screen Control, aptly enough) For those attempting to follow along with other documentation, a complete list of these sequences is shown in Figure 1.

As you can see, even when the ST isn't being a graphics machine, per se, it is still remarkably ST-like—powerful, versatile, and flexible. Though we haven't really even scratched the surface of the low-level functions of the ST in this brief discussion, we hope to have given you some inkling of the complexity and wealth of options available to the persistent experimenter.

Next issue, (drum roll, please), it's back up into the ozone layer of the AES for an introduction to GEM objects. ■

Listing 3.

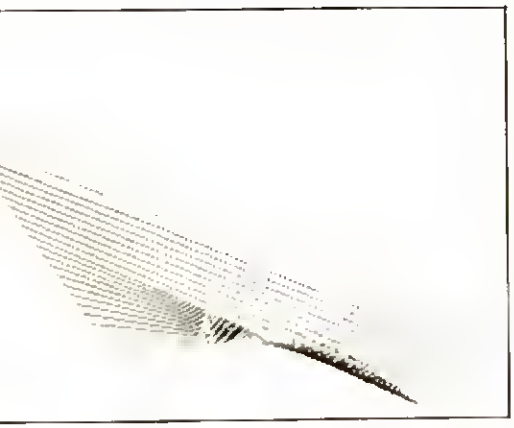
```
#include <osbind.h>

main()
{
        int rez;

        rez = Getrez();
        if (rez == 0) con_string("\33b\42\33c\57\33E");
        else if (rez == 1) con_string("\33b\42\33c\43\33E");
        exit(0);
}

con_string(string)
char *string;
{
        while(*string != '\0') Bconout(2,*string++);
}
```

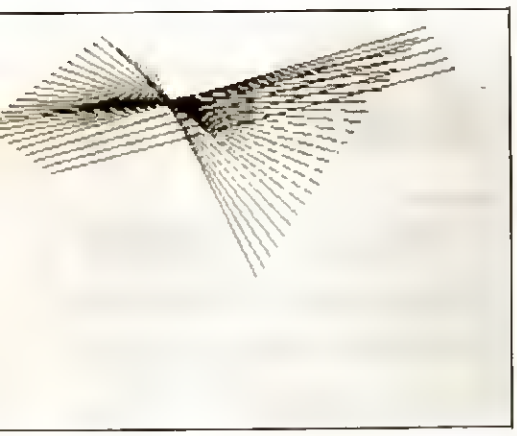
Figure 1. Screen control sequences.

<esc> A	Cursor up
<esc> B	Cursor down
<esc> C	Cursor right
<esc> D	Cursor left
<esc> E	Clear screen, home cursor
<esc> H	Home cursor
<esc> I	Reverse linefeed (vertical tab)
<esc> J	Erase from present cursor position to end of screen
<esc> K	Erase from present cursor position to end of line
<esc> L	Insert blank line at cursor
<esc> M	Delete line at cursor
<esc> Yrc	Put cursor at r,c where r,c are characters with codes (desired row + 32) and (desired column + 32)
<esc> bc	Set foreground color. The first c is simply the character c. The second c is e character with code (desired color index + 32).
<esc> cc	Set background color to c, where c is a character with code (desired color index + 32)
<esc> d	Erase from cursor position to start of screen
<esc> e	Make cursor visible
<esc> f	Make cursor invisible
<esc> l	Erase line at cursor
<esc> o	Erase line to left of cursor
<esc> p	Enter reverse video
<esc> q	Exit reverse video
<esc> v	Wrap text at end of line
<esc> w	Discard text at end of line

A hypnotic demonstration of high speed ST graphics

Helix





Remember the arcade game Qix? In Qix, the player builds walls across the playfield, striving to confine a whirling helix of lines in a smaller and smaller area. Though the game itself was challenging, the real selling point of Qix was the helical Qix itself: whirling about the screen, bouncing off the walls, stretching and twisting into iridescent, almost three-dimensional patterns.

To program a helical effect similar to that of Qix, you might begin by defining two endpoints, drawing a line between them, then moving the endpoints by adding or subtracting certain constant values, drawing another line, etc. Once a predetermined number of lines had been drawn—say, eight—you would go back to the first line and erase it. Then you would draw the ninth line and erase the second, draw the tenth and erase the third, and so on. By repeating this process, the helix could be made to move across the screen in a continuous wave.

Sounds simple enough—but hold on. Your program has to have some way of remembering where your line segments are so it can erase them later. What would happen if you tried storing the endpoints in successive elements of an array?

By JOHN JAINSCHIGG

ATARI KEY

- Any Atari ST Computer
- Mark Williams C-Language Development System

```
#include <osbind.h>

/*
   Setup: Declare control and data arrays required by AES/VDI,
   some workstation-related arrays and variables, and a pair
   of format strings for the program's alert boxes.
*/

int contrl[12],intin[128],ptsin[128],intout[128],ptsout[128];
int work_in[12],work_out[57],handle,xl,yl,cl;
char *clrs = "[0][ Do you prefer helixes in ...  ][COLOR|BLACK]";
char *cret = "[0][    Would you like to ...      ][CANCEL|RESTART]";

/*
   In main() we initialize the application, open a workstation,
   perform the lines routine until told to stop, close the
   workstation, and exit the application.
*/

main()
{
   appl_init();
   open_vdi();
   while(lines());
   v_clsvwk(handle);
   appl_exit();
}

/* lines() is the heart of the matter. It runs the line demo. */

lines()
{
   int v[4],p[120];
   int i,bound,stat,mx,my,chroma = 0;
   int p1 = 0,p2 = 4,q = 4;

/*
   We start by clearing the point-storage array, and setting
   start coordinates for our two points at random. We also set
   vectors of motion (x and y-deltas) with random magnitudes.
*/

   for (i = 4; i < 120; p[i++] = 0);
   p[0] = rand(xl); p[1] = rand(yl);
   p[2] = rand(xl); p[3] = rand(yl);
   for(i = 0;i < 4;v[i++] = rand(4) + 2);

/*
   If we're running in monochrome, we don't bother asking the
   user if he/she wants color in the display. Otherwise we do.
   Variable cl = color range in present graphics mode.
*/

   if (cl > 1){
      if(form_alert(2,clrs) == 1) chroma = cl;
      }

/* Ditch the mouse-cursor, clear the screen, and off we go. */

v_hide_c(handle);
v_clrwk(handle);
```

```
/*
   Select a random color (in color modes) and draw a line.
   Undraw another line. In the beginning, of course, these
   "other lines" haven't yet been initialized, so what we're
   really doing is running our second pointer once around an
   array filled with zeros, undrawing a point in the upper
   left hand corner of the screen. A reprehensible dodge, but
   it eliminates an "are we ready to start undrawing lines yet?"
   flag and test in the loop.
*/
do{
   vsl_color(handle,rand(chroma)+1);
   v_pline(handle,2,&p[p1]);
   vsl_color(handle,0);
   v_pline(handle,2,&p[p2]);

/*
   Do screen boundary checks, negate appropriate movement
   vectors if helix is about to run off the screen, add
   movement vectors to present endpoints to generate new ones.
   while() loop makes it possible to use the results of our
   test addition as our new value except when we hit a border.
*/

   for (i = 0;i < 4;i++){
     if (i == 0 || i == 2) bound = xl;
     else bound = yl;
     while((p[q + i] = p[p1 + i] + v[i]) > bound ||
           p[q + i] < 0) v[i] = -v[i];
     }

/* Increment our pointers and wrap around as necessary */

   p1 = (p1 + 4) % 120;
   p2 = (p2 + 4) % 120;
   q = (q + 4) % 120;

/* Any wild mouse activity? If not, continue. */

   vq_mouse(handle,&stat,&mx,&my);
   } while (stat == 0);

/* Clear screen, show mouse, ask user what he/she wants. */

   v_clrwk(handle);
   v_show_c(handle);
   return(form_alert(2,cret) - 1);
}

/* Open a virtual workstation; get bounds and color range. */

open_vdi()
{
   int i,d;

   for(i=0;i<10;work_in[i++]=1); work_in[10]=2;
   handle = graf_handle(&d,&d,&d,&d);
   v_opnvwk(work_in,&handle,work_out);
   xl = work_out[0]; yl = work_out[1];
   cl = (xl / yl) + 14 * (xl == 319);
}

/* Get random number between 0 and x-1 */

rand(x)
int x;
{
   return(Random() % x);
}
```

Suppose you started one index off at zero, drew a line, stored the endpoints at the indexed position in the array, and incremented the pointer. Once eight lines had been drawn, you would start a second pointer off at zero, grab the endpoints stored there, undraw the line between them, increment both pointers, and continue. As the helix moved, successively higher and higher ranges of array elements would be accessed by the pointers. Eventually you would run out of array space, at which point your helix would run out of gas.

The solution to this problem is apparent when you realize that drawing a helix of eight lines means keeping track of only eight sets of endpoints at a time. Each time you "undraw" a line, the array element containing its endpoints is freed up and can be used again. You can turn this fact to your advantage by creating a data structure called a *ring buffer*—an array that can be accessed in circular fashion by means of "wrap-around" pointers.

You start by setting up an array with only eight elements, and setting your first index to zero, as before. You draw your first line, store its endpoints in the indexed element, and increment the index. When you reach the eighth line, you start your second index off at zero, grab the endpoints stored there, undraw the line between them, and increment both indices, as you did before.

The top index now points beyond the end of your array, so you wrap it around to zero once again (using a test-and-branch or a modulo instruction) so that it points to the newly-vacant lowest element, causing the endpoints of "line 9" to be stored there. You can continue this process indefinitely, wrapping the pointers around one after the other, and never run out of array space.

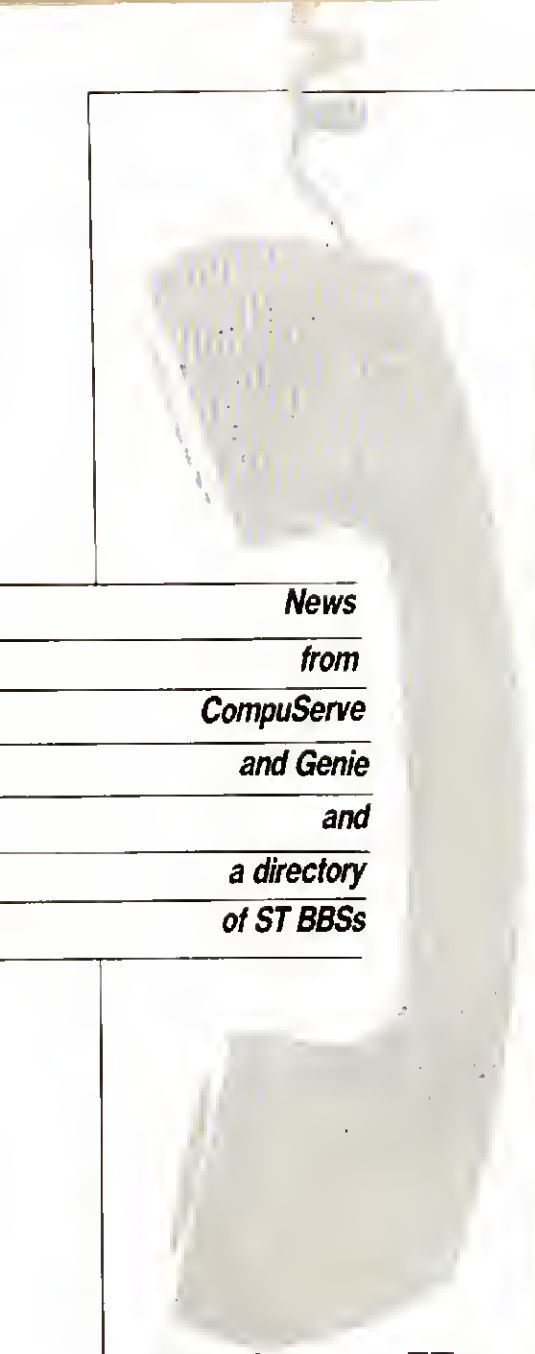
The program Helix, presented here written in Mark Williams C, uses the ring buffer approach to create an extremely high speed, highly variable, multicolor helix effect. After you double-click on the program icon, an alert box asks you to choose whether you prefer your helixes in color or black. The program then chooses random direction and speed parameters for the current helix and sets it in motion.

You will find that each helix has its own personality, and that a single helix can be observed for quite a while before repeating its characteristic pattern of motion. When you have tired of a particular helix, pressing either mouse button will bring up a second alert box, asking you if you care to exit the program or restart the effect with new random parameters.

To run Helix, enter the listing as shown into a file called helix.c, using MicroEMACS or an equivalent editor. When you are finished, compile the program from the Mark Williams MSH shell $ prompt by typing:

```
$ cc helix.c -laes -lvdi
```

The -laes and -lvdi switches serve to link in the necessary Application Environment Services and Virtual Device Interface libraries. ■



News from CompuServe and Genie and a directory of ST BBSs

Teletalk

By JOHN J. ANDERSON

Howdy-do, modemers. I trust your carrier tones are strong and steady, and that handshaking is enabled. Let's have a look online and see what's new.

BBS Update

The first item of business this month is the Atari ST BBS Hotlist, Edition 3.0, courtesy of The Hotline BBS in Arlington, VA, which appears as Table 1.

New on CompuServe

If you have read my review of *Aegis Animator ST* elsewhere in this magazine, you already know how impressed I

Table 1. Atari ST BBS Hotlist.

St	BBS Name	Phone Number	Baud	Remarks
AK	Hole in the Wall	907-338-0348	3/12	8-Bit System
AL	Bloom County	205-772-8526	3/12	FoReM ST
AR	LRBBS	501-224-9454	3/12	ST-Unk Software
AZ	Phase Computing BBS	602-258-7205	3/12	ST Citadel
AZ	Bookman's ST Emporium	602-745-2837	3/12	ST-Unk Software
CA	Fresno ST User Group	209-292-1777	3/12	MichTron ST
CA	Mount Olympus	209-526-9889	3/12	FoReM ST
CA	Weird City*	213-273-5234	3/12	FoReM ST
CA	PC Heaven*	213-374-7929	3/12	FoReM ST
CA	MIDI World Network*	213-826-4288	3/12	MichTron ST
CA	The Mid-Cities Connection*	213-867-1943	3/12	MichTron ST
CA	Micronic Connection*	213-867-5073	3/12	MichTron ST
CA	Yamo's Yacking Yard	408-227-7868	3/12	MichTron ST
CA	Star Fleet Academy	408-244-1614	3/12/24	BB/ST
CA	Atari Shoppe BBS	408-249-4029	3/12	ST-Unk Software
CA	Iron Works	408-257-7147	3/12/24	ST-Unk Software
CA	X-Factor	408-279-8086	3/12	ST-Unk Software
CA	Ace Connection	408-353-4531	3/12	BB/ST
CA	B.A.U.G.	408-358-1520	3/12	MichTron ST
CA	OSS, Inc.	408-446-3451	3/12	8-Bit System
CA	Atari Corporation #1	408-745-5308	3/12	MichTron ST
CA	Atari Corporation #2	408-745-5970	3/12	MichTron ST
CA	Atari Corporation #3	408-745-2642	3/12	MichTron ST
CA	Atari Corporation #4	408-745-4758	3/12	MichTron ST
CA	Atari Corporation #5	408-745-5664	3/12	MichTron ST
CA	SJC Computer	408-778-0205	3/12	FoReM ST
CA	KOME	408-985-8875	3/12	MichTron ST
CA	ST Users BBS*	415-221-9505	3/12	MichTron ST
CA	Beckemeyer Development	415-452-4792	3/12	FoReM ST
CA	WelchNet*	415-664-2811	3/12	Other System
CA	SLCC BBS	415-782-4402	3/12	ST-Unk Software
CA	Liad BBS	415-939-3586	3/12	ST-Unk Software
CA	Castle BBS	415-939-5606	3/12	ST-Unk Software
CA	STrategic Command	415-939-6275	3/12	MichTron ST
CA	Amber BBS	415-964-0584	3/12	ST-Unk Software
CA	Atari Line ExPgi's	619-282-6608	3/12	ST-Unk Software
CA	Stallion BBS	619-428-0319	3/12	ST-Unk Software
CA	Ridgecrest BBS	619-375-6750	3/12	FoReM ST
CA	ST*SIG Online Edition	619-726-4419	3/12	MichTron ST
CA	SMART	619-741-2369	3/12	ST-Unk Software
CA	STATE (ST Applications Mag.)	707-585-2194	3/12	8-Bit System
CA	Pipeline	805-526-5660	3/12	8-Bit System
CA	Camp Crystal Lake	818-352-0046	3/12	ST-Unk Software
CA	Inner Realm	818-352-7102	3/12	FoReM ST
CA	The Way BBS	818-363-3505	3/12	Other System
CA	Infinity	818-760-0943	3/12	FoReM ST
CA	Exceptional Software BBS	818-769-9538	3/12	ST-Unk Software
CA	Atari 16+32	818-996-9565	3/12	ST-Unk Software
CO	Electric Land*	303-447-2227	3/12	Other System
CO	Mycroft*	303-699-7195	3/12	Other System
CO	ForeST of Mordor*	303-972-8566	3/12	FoReM ST
CT	Pyrotechnics BBS	203-287-0780	3/12	ST-Unk Software
CT	Skyline BBS	203-488-0816	3/12/24	FoReM ST
CT	Excalibur	203-776-1411	3/12	ST-Unk Software
CT	Hi-Tech BBS	203-776-2850	3/12/24	FoReM ST
DE	Delaware BBS	302-378-2277	3/12	MichTron ST
FL	ST Repeater	305-247-7789	3/12	MichTron ST
FL	Computer Spectrum	305-251-1925	3/12	MichTron ST
FL	Southern Atari Remote	305-266-6178	3/12	MichTron ST
FL	SST BBS (SST Systems)	305-383-0353	3/12	ST-Unk Software

St	BBS Name	Phone Number	Baud	Remarks
FL	National Users Group BBS	305-383-1413	3/12	ST-Unk Software
FL	Battlestar Atarian	305-488-0480	3/12	FoReM ST
FL	Police Department	305-574-9483	3/12/24	FoReM ST
FL	A.C.C.P.B.	305-585-7879	3/12	ST-Unk Software
FL	The STarship BBS	305-658-1425	3/12	MichTron ST
FL	The Disk Bank	305-774-8986	3/12/24	ST-Unk Software
FL	LBBS	305-788-6226	3/12	8-Bit System
FL	A.S.U.G.	305-793-9385	3/1	ST-Unk Software
FL	McDonald Development Group	305-886-1632	3/12	MichTron ST
FL	McDonald Computer Center	305-896-6707	3/12	MichTron ST
FL	Computer Chip BBS	813-351-3604	3/12/24	Other System
FL	Sota Computers	813-924-4590	3/12	Other System
GA	BaSTille BBS*	404-627-1995	3/12	MichTron ST
HI	Maximum Security	808-422-2247	3/12	MichTron ST
ID	Hig's Hideout	208-233-7742	3/12	MichTron ST
ID	Atari Kingdom	208-522-3583	3/12	FoReM ST
IL	JAF Data Systems BBS	312-238-4328	3/12/24	Other System
IL	Blue Moon*	312-457-2219	3/12	MichTron ST
IL	New Dawn*	312-459-8498	3/12	MichTron ST
IL	SCAT BBS*	312-462-9844	3/12	8-Bit System
IL	The Remote Atari BBS*	312-587-1349	3/12	ST-Unk Software
IL	Dark Side of the Moon*	312-639-1993	3/12	FoReM ST
IL	Back Door*	312-725-6232	3/12	MichTron ST
IL	SunDog BBS*	312-759-9101	3/12	ST-Unk Software
IL	ChicagoLand AUG	312-889-1240	3/12	Other System
IL	Stash BBS	312-980-5617	3/12	MichTron ST
IL	Porthole to Oblivion	815-838-3615	3/12	ST-Unk Software
IN	Hart City BBS	219-262-3980	3/12	FoReM ST
IN	Megahertz	812-379-1162	3/12	FoReM ST
KS	KC Atari	816-353-7541	3/12	ST-Unk Software
KS	Bloom County	913-842-5011	3/12/24	Other System
LA	Outlet's BBS	504-277-3541	3/12	ST-Unk Software
LA	Bit Map BBS	504-394-6224	3/12	MichTron ST
LA	American Networks III	504-436-7222	3/12/24	Multi-line
MA	Games People Play	617-471-2270	3/12	MichTron ST
MA	Metro ST	617-536-6974	3/12	MichTron
MA	FoReM ST (Commnet Systems)	617-620-0903	3/12/24	FoReM ST
MA	CompuClub BBS	617-788-0024	3/12	Other System
MA	Freeport Express*	617-825-3124	3/12	Other System
MA	Harbor BBS	617-929-8678	3/12	MichTron ST
MD	Tanj BBS*	301-251-0675	3/12	8-Bit System
MD	Mission Impossible*	301-384-6153	3/12	FoReM ST
MD	Blackbird BBS*	301-445-5496	3/12/24	FoReM ST
MD	The 8th Dimension*	301-460-6030	3/12	MichTron ST
MD	The Pentagon*	301-464-0948	3/12	8-Bit System
MD	Vanguard BBS*	301-577-3569	3/12	MichTron ST
MD	ST Hospital*	301-654-0565	3/12/24	FoReM ST
MD	Cal Com BBS*	301-681-8933	3/12	Other System
MD	Starship	301-833-4361	3/12	MichTron ST
ME	Maniac	207-854-2687	3/12	MichTron ST
MI	Griffon BBS (MichTron)	313-332-5452	3/12	MichTron ST
MI	Club II BBS	313-334-8877	3/12	ST-Unk Software
MI	Surfboard BBS*	313-547-5671	3/12	FoReM XE
MI	Ann Arbor BBS	313-665-7286	3	ST-Unk Software
MI	Tony's Corner/Collosus BBS*	313-754-1131	3/12	ST-Unk Software
MI	MACE BBS	313-978-1685	3/12	8-Bit System
MI	SuperServe	616-791-2109	3/12	ST-Unk Software
MN	Spike Master	612-374-3232	3/12	FoReM ST
MN	STatus	612-777-6376	3/12	MichTron ST
MO	Parthanon	314-522-1460	3/12	FoReM ST
MO	A.U.R.A.	314-928-0598	3/12	FoReM ST
MO	KC Atari Central	816-353-7541	3/12	FoReM ST
MO	Rampart General	816-356-6142	3/12/24	FoReM ST
MO	Atari Outpost	816-765-1150	3/12	FoReM ST
MS	Coastal Area BBS	601-388-3490	3/12	MichTron ST
NC	East Coast Systems	704-529-0375	3/12	FoReM ST
NE	Atari-O!	402-592-4435	3/12/24	FoReM ST
NJ	The Border BBS*	201-549-8419	3/12	MichTron ST

was with the program. If you are interested in seeing the program in action, head over to DL10 of the Atari 16-Bit SIG. There you can find HEADRO.ARC/binary, an animation made with the program. It features that total star of the tube, Max Headroom.

You will need to capture PLAYER-.ARC/binary to view the animation. That is the stand-alone player for *Animator ST*. Make sure that it resides in the same directory as the animation you want to view.

If you are interested in the possibility of purchasing *Animator ST*, this file will give you an idea of the capabilities of the program. It is also likely that future animations will appear as uploads to the ST SIG, so hold on to your copy of the player.

New in DL1 is CHECKE.PRG/binary, a neat little checkers program that runs in medium resolution only. The program is in the public domain.

Also to be found there is SPACEW.ARC/binary, a fairly good version of Space War (like the old Atari Coin-Op game). It is strictly keyboard controlled and requires two players. The mono mode is more finely tuned, but the color has nice explosions. Takes about 11 minutes to download at 1200 baud.

If you would like to have a look at Michtron's *Time Bandits*, download TIMEBA.ARC/binary from DL1. It is version 0.96 of the game that the folks a Michtron have been kind enough to put into the Public Domain. It is not nearly as good as the current version (2.1), but it should whet your appetite for the real thing, as it is well worth the download time (618 blocks Xmodem).

In DL3 you will find TD.ARC/binary, a time and date management utility that simplifies entering at boot time. It also will use the DS1216E SmartWatch, the chip used in most clock kits, if one is installed.

Alongside that program in DL3 is TURTLE.ARC/binary, which will be of interest to hard disk owners. This is version 2.11, replacing 2.10 (which had a bad resource file). The hard disk backup program supports TWISTER format disks, recovers from hard disk read errors, eliminates the need for resets, and is far less vulnerable to folder crashes. If you use Supra's autoboot, run the proper RAMdisk yourself before running TURTLE. The autoboot sets all the bits in the drive map, so all drives appear to

be active. See CHANGES in the ARC file for full details.

New on Genie

Desk accessories are among the handiest items around for your Atari ST, and Library 1 of the Genie ST Roundtable is chock full of the little biters. Let's have a look around.

STARTUPM.ARC displays the free memory, time, and date in the upper right corner of the ST screen and inserts an entry in the desk menu titled Other. From Other you can run a TOS program, idle the computer (blank the screen with an occasional ST drawing) or reset the desktop parameters from a desktop.inf file. A nice little accessory.

EZFRMT14.ARC is a disk formatter program. It is very easy to use and always there when you need it. It does regular ST format plus the extended format of 10 sectors per track, 82 tracks. It also has error trapping (of sorts).

RAMBUFFR.ARC is a combination RAMdisk and printer buffer that is accessible from within any GEM-based program. The documentation is ARC ed along with the program files. Very nicely done.

CLOCKACC.ARC is an analog clock desk accessory. This one is a lot simpler than the one from Atari and has no cute arrows on the end. C source included.

TINYTOOL.ARC is a great little desk accessory for every programmer. It is a memory, sector, or file editor that can be accessed as an accessory from the desktop or from within any GEM-based program.

MOBZUTIL.ARC is a file utilities program that is installed as a desktop accessory upon booting. It allows you to copy, delete, create folders, rename, check memory, etc.

ALARMCLK.ARC is a desktop accessory courtesy of Micro-Time Electronics. It allows you to set up to five different alarm reminders to go off from within any GEM application. It also places a digital clock in the top right corner of the screen. The utility supports "?" wild cards, so you can have the alarms armed to go off hourly/daily/monthly, etc. It allows for a reminder message of about 35 characters for each alarm. Alarms can be saved to a .DAT file for regular use and also disabled temporarily from within any GEM program. The ARCed file contains documentation, and the program works in all

St	BBS Name	Phone Number	Baud	Remarks
NJ	Fizben's Towers	201-583-5758	3/12	MichTron ST
NJ	Dark Side	201-782-3727	3/12/24	ST-Unk Software
NJ	AtlantIST	201-793-0996	3/12	MichTron ST
NJ	Backstage BBS*	201-944-1996	3/12	ST-Unk Software
NJ	ST Wizard	609-627-4556	3/12	FoReM ST
NJ	Tannoy Control	609-953-8496	3/12	MichTron ST
NV	520 ST Systems	702-363-8111	3/12	ST-Unk Software
NV	Rebel BBS	702-435-0786	3/12	Other System
NV	Computer World	702-735-7264	3/12	Other System
NV	The Shelter	702-826-9633	3/12	FoReM ST
NY	Fever ST BBS*	212-562-5161	3/12	MichTron ST
NY	QMI BBS (QMI, Inc)	315-457-7216	3/12	BB/ST
NY	YCAMP	516-295-0823	3/12	FoReM ST
NY	Strike Force	516-368-2476	3/12	FoReM ST
NY	ST Realm	516-536-1068	3/12/24	ST-Unk Software
NY	Computer Palace	516-698-6182	3/12	MichTron ST
NY	Atari Link	516-924-1281	3/12/24	FoReM ST
NY	Ateri Apex BBS	716-458-2638	3/12/24	FoReM PC
NY	Crazy Treln	716-741-9822	3/12	ST-Unk Software
NY	Army Outpost	718-332-5851	3/12	FoReM ST
NY	World of Krypton	718-373-0623	3/12	ST-Unk Software
NY	IMF BBS	718-476-9659	3/12	8-Bit System
NY	Dateline (B.A.S.I.C.)	718-648-0947	3/12	MichTron ST
NY	Imperium Galactic	718-851-5785	3/12/24	FoReM ST
NY	Beehive BBS	718-852-2823	3/12	FoReM ST
NY	Hunter's Hangout	718-897-5578	3/12	FoReM ST
NY	The Telephone Company	817-778-2506	3/12	MichTron ST
OH	Pep Board	218-888-5466	3/12	FoReM ST
OH	Flag City ST	419-423-0206	3/12	FoReM ST
OH	Powerhouse	419-472-6835	3/12	FoReM ST
OH	InfoNet	513-435-8381	3/12	Other System
OH	The Scan Line	513-829-0389	3/12	MichTron ST
OH	The Pixel Palece	513-856-9742	3/12	ST-Unk Software
OK	M*A*S*H	405-743-0651	3/12	MichTron ST
OR	Atarlen	503-245-9730	3/12	8-Bit System
OR	I.B. Computers	503-292-1321	3/12	FoReM ST
OR	A.C.E. of Eugene	503-343-4352	3/12/24	8-Bit System
OR	SS BBS	503-479-9516	3/12	Other System
OR	The Machine	503-747-8758	3/12	8-Bit System
OR	Data Dogs BBS	503-935-4605	3/12	FoReM ST
PA	ST Atari Haven*	215-586-8705	3/12	FoReM ST
PA	ST Emporium*	215-757-7308	3/12/24	FoReM ST
PA	Cornerstone BBS	215-775-5684	3/12	MichTron ST
PA	Intentional Intellect BBS	215-791-2304	3/12/24	FoReM ST
PA	Atari Elite	412-384-5609	3/12	ST-Unk Software
PA	The Underground BBS	717-735-8885	3/12	FoReM ST
RI	Narragansett Bay BBS	401-683-9115	3/12	FoReM ST
SC	Springdale BBS	803-794-3307	3/12	ST-Unk Software
SD	M.U.G. BBS	605-226-1590	3/12	MichTron ST
TN	Troll's Cave	615-242-7909	3/12/24	FoReM ST
TN	Computerama BBS	615-892-1454	3/12	ST-Unk Software
TX	Principle Oiscordle ST	512-444-1840	3/12	MichTron ST
TX	Trade/Sell BBS of Texas*	713-242-6041	3/12	Other System
TX	Computers to Grow BBS*	713-271-0571	3/12/24	Other System
TX	Atari ST*	713-541-9160	3/12	MichTron ST
TX	Imagery*	713-928-2392	3/12	MichTron ST
TX	HASTE*	713-955-9532	3/12/24	MichTron ST
TX	ST BBS*	713-995-0843	3/12	ST-Unk Software
TX	Dal-Ace BBS	817-267-4913	3/12	ST-Privete BBS
TX	Church Mouse BBS	817-284-5725	3/12	MichTron ST
TX	Jenkins Computer Store (STEP)	915-751-7837	3/12/24	ST-Unk Software
UT	Atari Connection	801-377-1617	3/12	ST-Unk Software
UT	Well of Souls	801-485-6349	3/12	FoReM ST
UT	Robotech	801-561-9500	3/12	MichTron ST
UT	Weber State College RBBS #2	801-626-7906	3/12	Other System
UT	Matrix	801-628-5755	3/12	FoReM ST
UT	Wizard's Workshop	801-789-8439	3/12/24	FoReM ST
UT	Heart of Gold	801-964-9764	3/12	FoReM ST

St	BBS Name	Phone Number	Baud	Remarks
VA	Gallifrey*	703-256-3223	3/12	FoReM ST
VA	Rivendell*	703-437-9380	3/12	FoReM ST
VA	The Other Side of Reality*	703-451-4412	3/12	FoReM ST
VA	WAACE ST (Novatari)*	703-569-3227	3/12	FoReM ST/Fee
VA	The Hotline/Hotlist Author*	703-683-3944	3/12	FoReM PC
WA	Bytes & Pieces*	206-241-8963	3/12	MichTron ST
WA	The Farm*	206-264-4281	3/12	MichTron ST
WA	Foundation & Empire BBS*	206-323-1330	3/12	FoReM ST
WA	Asylum BBS	206-347-1008	3/12	FoReM ST
WA	ST Bit Bucket*	206-363-8592	3/12	FoReM ST
WA	New World*	206-365-6938	3/12	MichTron ST
WA	DOS File ST BBS*	206-367-3453	3/12	ST-Unk Software
WA	Suburbia 1200*	206-488-7496	3/12	FoReM ST
WA	M.O.T.	206-535-0574	3/12	FoReM ST
WA	SIG 16	206-636-3403	3/12	ST-Unk Software
WA	Bauhaus BBS*	206-672-0956	3/12	FoReM ST
WA	Xanth BBS*	206-682-8039	3/12/24	FoReM ST
WA	Xanth BBS #2*	206-823-9707	3/12/24	FoReM ST
WA	Space Butlers*	206-941-2824	3/12	FoReM ST
WI	After Hours BBS (7pm-8am)	414-235-9164	3/12	MichTron ST
WI	Atari Havoc	414-545-6230	3/12	FoReM ST

Canadian ST BBSs

68000 Mice	403-242-0706	3/12	FoReM ST
Northern Lights	403-791-0457	3/12	8Pm-3Pm
Megabaud	416-243-9519	3/12	MichTron ST
ST Connection	416-276-4101	3/12	FoReM ST
Atari Corp of Canada	416-579-2169	3/12	MichTron ST
ST BBS	416-662-7717	3/12	MichTron ST
Gold Board (12pm-8pm)	416-691-4859	3/12	ST-Unk Software
Starlog	416-926-8874	3/12	FoReM ST
FASTER	514-489-0680	3/12	ST-Unk Software
ST Base One	514-671-5719	3/12	ST-Unk Software
S.T.A.R.S.	519-472-1638	3/12	FoReM ST
Natl Capital Atari Users	613-231-3411	3/12/24	FoReM ST
The Grue's Lair	613-394-3676	3/12	MichTron ST
Ali Computer	613-745-6372	3/12	ST-Unk Software
Media Land	613-820-7326	3/12	FoReM ST
Husky BB9Tv9	705-949-7366	3/12	ST-Unk Software

Note: This list is provided as a service for *Atari Explorer* readers. The magazine does not endorse any of the BBSs listed here nor does it make any judgment as to the legality or appropriateness of their content.

*Denotes that the BBS is accessible via Telenet's PC Pursuit.

PC Pursuit Info: (Data)800-835-3001; (Date) in VA 703-689-2987; (Voice)800-TELENET 8:00 a.m.-5:00 p.m. Eastern Time. BBSs in Maryland (MD) and Virginia (VA) marked as being accessible through PCP can be reached through the Washington DC (202) node.

resolutions.

KALKLOCK.ARC is a very nice Clock/Calendar. The ARC file contains full documentation .ACC and .PRG versions of the program. The calendar can display any month between years 1753 and 2399. It can also display the number of days between two dates. The clock can be set through the .ACC version of the program.

NUINTRAM.ARC is an improved version of the INTERSECT Installable/De-installable RAMdisk. It allows you to set it up for auto install at boot-up and still de-install and re-install it at will. Also included is documentation that will allow you to use it in a snap.

DIRPRINT.ACC is a desk accessory to print your disk directories to your printer. The formatting of the printout isn't all that great, but it does do the job. Just put this file on the disk you use for boot-up, and it will be there when you need it.

ACCLOADER.PRG is very neat. Put it in an auto folder on your boot disk, and you can then have as many (or as few) desk accessories as you want. When loaded, choose the accessories you want; the program automatically names all unwanted desk accessories with an .ACX extension. Easy to use, and very handy. ■

INDEX TO ADVERTISERS

User Groups!

Don't miss this opportunity to tell the world about your user group. See User Friendly in this issue for details.

Part 3: A nuts and bolts guide to MIDI

Sound Chip

By JOHN JAINSCHIGG

Last time, we covered MIDI Note On and Note Off messages, learned something about running status, and made a few stabs at analyzing incoming MIDI information. This issue, in an incredible flurry of hasty explanations and intellectual prestidigitation, we're going to examine the rest of the MIDI protocol and hack together a powerful program to disassemble an incoming MIDI data stream. Fasten your seat belt!

Table 1 covers the whole MIDI protocol. Memorize it. The end.

Actually, I'm just kidding. If you feel like memorizing Table 1, which indeed does cover the whole of the MIDI protocol, go right ahead —it's probably good for the soul. But whether you memorize it or not, this is just the beginning. Now it's intellectual overview time.

Channel Voice Messages

Channel Voice messages transmit data about performance actions and dynamics. Besides *Note On* and *Note Off* (the sine qua non of la musique), the most important Channel Voice messages are Program Change, Pitch Wheel Change, and Control Change.

Program Change is sent when you press one of your preset buttons, changing the sound produced by your instrument or carrying out some other preprogrammed action. The protocol supports up to 127 different programs per channel.

Pitch Wheel Change is sent each time your pitch bend wheel or slider moves. The two-byte value sent along with Pitch Wheel Change deseribes the momentary position of the wheel relative to neutral (0). Turning the wheel all the way up thus produces a veritable flurry of messages—up to 64 separate messages on a Casio CZ-101, which uses only the lower significant byte to transmit wheel position. More sophisticated synths use both bytes, dividing the are of the turning of the wheel into up to 16,129 positions and sending thousands of messages each time it moves.

Note that values sent by Pitch Wheel Change describe only the position of the wheel and bear no absolute relation to how much notes are actually bent. The portamento range controlled by the pitch wheel is variable and set independently for each instrument in a MIDI chain. Thus, when one synth is acting as "master" to another, for example, moving the pitch wheel on the master synth may bend its sounding notes upward by a fifth, while the same notes sounding on the slave synth are bent up by only a third.

With Program Change looking after the patch banks and Pitch Wheel Change handling the bend wheel, the job of *Control Change* is to handle, well, everything else. Each of the other knobs, dials, pedals, buttons, levers, and gizmos on the synthesizer control panel is keyed to one of the 122 different "control" values that can be sent with Control Change.

The state/position/setting of the control in question is sent in the second data byte. The logieal arrangement of controls and the rules governing the formulation of Control Change messages about them is complex, and we'll leave off discussing it for now.

Polyphonic Key Pressure messages are sent by sophisticated synthesizers capable of distinguishing how hard an individual key is being held down. *Channel Pressure* messages are sent by less sophisticated synths and reflect the pressure on the key that is being held down hardest; when received by a slaved synth, this value controls the sound of all notes in the given channel. Inexpensive synths like the Casio don't have pressure-sensitive keyboards, hence don't send either of these message types.

Channel Mode Messages

Channel Mode messages, as you can see by looking at their command byte, actually form a subset of Control Change messages. They come in six flavors.

Local Control messages revector the sound generating facilities of the synth. When Local Control Off is received, a synthesizer produces sound only in response to incoming MIDI data, while continuing to generate MIDI data in normal fashion. Sound control is revectored to the instrument controls when Local Control On is received.

All Notes Off is a special message used to shut down sounding notes on all channels. It is used as a sort of panic button.

Omni Off and *On*, *Mono*, and *Poly* control the ability of the synth to respond to note data on multiple channels and control allocation of voiees. Four combinations of the commands can be formed, each with different effects.

	Status Byte High Nibble	Low Nibble	Data Bytes	
CHANNEL VOICE MESSAGES				
Note Off	1000	Channel: 0-15	Note: 0-127	Velocity: 0-127
Note On	1001	Channel: 0-15	Note: 0-127	Velocity: 0-127
Polyphonic Key Pressure	1010	Channel: 0-15	Note: 0-127	Pressure: 0-127
Control Change	1011	Channel: 0-15	Ctrl Number: 0-121	Value: 0-127
Program Change	1100	Channel: 0-15	Program: 0-127	
Channel Pressure	1101	Channel: 0-15	Pressure: 0-127	
Pitch Wheel Change	1110	Channel: 0-15	LSB: 0-127	MSB: 0-127
CHANNEL MODE MESSAGES				
Channel Mode Change	1011	Channel: 0-15	Local Control: 122	On: 0
				Off: 127
			All Notes Off: 123	Dummy: 0
			Omni Mode Off: 124	Dummy: 0
			Omni Mode On : 125	Dummy: 0
			Mono Mode On : 126	# Voices in Receiver
			Poly Mode On : 127	Dummy: 0
SYSTEM COMMON MESSAGES				
Undefined	1111	0001		
Song Position Pointer	1111	0010	LSB: 0-127	MSB: 0-127
Song Select	1111	0011	Song: 0-127	
Undefined	1111	0100		
Undefined	1111	0101		
Tune Request	1111	0110		
End-of-Exclusive	1111	0111		
SYSTEM REAL-TIME MESSAGES				
MIDI Clock	1111	1000	(24 beats per quarter note)	
Undefined	1111	1001		
Start Sequence	1111	1010		
Continue Sequence	1111	1011		
Stop Sequence	1111	1101		
Active Sensing	1111	1110	(Send once/300ms. when no other activity)	
System Reset	1111	1111		
SYSTEM EXCLUSIVE MESSAGES				
Whatever	1111	0000	Manuf. ID: 0-127 (Followed by data)	
			End with EOX (11110111)	

Table 1. MIDI Byte Table.

Omni On/Poly On causes the synth to accept voice messages regardless of channel and assign them to voices as they come in. All voice messages produced by the synth are sent out under the selected basic channel.

Omni On/Mono On permits the synth to accept voice messages on any channel, but permits only one voice to sound at a given time. Likewise, only a single voice can be produced at the keyboard, and MIDI messages for this voice are sent out under the selected basic channel.

Omni Off/Poly On causes the synth to ignore voice messages on any channel except the selected basic channel. Messages received on this channel are assigned polyphonically to available voices as they come in. Voice messages produced by the synth are sent out under the basic channel.

Omni Off/Mono On causes the synth to accept a given number of voices N on a sequence of channels starting with the basic channel and assign each of these, monophonically, to voices 1 through N-1. Voice messages produced at the keyboard are assigned monophonically to channels between the basic channel and the basic channel+N-1.

System Common Messages

System Common messages are messages received and acted upon (as appropriate) by all equipment in a MIDI network, particularly master equipment such as sequencers. *Song Position Pointer* is used to set sequencers to a particular position in a song before commencing playback. It relays an offset of MIDI clock beats (24 of which occur each quarter note) into the song.

Song Select is used to tell multisequencers which song is now going to be played. *Tune Request* tells all analog synthesizers in the network to tune their oscillators to a common reference tone. The *End Of Exclusive* message informs the network that a System Exclusive data dump has been completed.

System Realtime Messages

System Realtime messages are used for synchronizing all the equipment on the MIDI network. The *Clock* message is sent out 24 times per quarter note by the master sequencer or clocking unit and serves as a timing reference for playback by subservient sequencers.

The *Start* message tells sequencers to commence playback at the beginning of the present song and causes drum machines and similar equipment to begin playing their patterns. The *Stop* message, naturally enough, does the opposite, halting the present sequence and shutting down auxiliary equipment.

The *Continue* message is used in combination with Song Position Pointer to restart playback at a particular point in a sequence. The *Active Sensing* message is optional; if it is ever received, equipment will expect to continue receiving it at intervals of no more than 300 milliseconds whenever the MIDI network is otherwise idle. If the signal fails to reoccur within this interval, equipment will shut down and assume its default state.

The *System Reset* message is another panic button. When received, it shuts the network down, putting everything back at power-up status.

System Exclusive Messages

Or, rather, the System Exclusive message—or SYSEX—is used to signify the start of a message intended for only one brand/model of equipment in

MIDI ANALYST

ATARI KEY

- Any Atari ST Computer
- ST Basic

```
10    dim t$(29,3),b(29),buf(9):bc=0:p=0
:rs=0
20    for i=0 to 28:read a:b(i)=a-1
30    for j=1 to a:read t$(i,j):next:nex
t
40    fullw 2:clearw 2
50    if inp(-3) then a=inp(3):goto 50
60    if not inp(-3) then 60
70    bv=inp(3) and 255
80    if (bv and 128)=0 then 150
90    if (bv and 240)=240 then 120
100   com=(bv and 112)/16:chn=bv and 15
110   bc=b(com):goto 60
120   if (bv and 8) then 140
130   rs=0:com=(bv and 15):bc=b(com+13):
goto 60
140   com=(bv and 7):bc=b(com+20):goto 6
0
150   buf(p)=bv:bc=bc-1:p=p+1
160   if bc>0 then 60
170   if com=3 and buf(0)>122 then com=7
+buf(0)-122
180   print t$(com,1);"-----";
190   if chn<13 then print "channel ";ch
n;"-----";
200   for i=0 to p-1
210   print t$(com,i+2);":";buf(i);" ";:
next i:print
220   p=0:if rs then bc=b(com)
230   goto 60
1000  data 3,note off,note,velocity
1010  data 3,note on,note,velocity
1020  data 3,poly pressure,note,pressure
1030  data 3,control change,control,sett
ing
1040  data 2,program change,program
1050  data 2,chan pressure,pressure
1060  data 3,pitch wheel,lsb,msb
1070  data 3,mode change,local control,o
n/off
1080  data 3,mode change,all notes off,d
ummy
1090  data 3,mode change,omni off,dummy
1100  data 3,mode change,omni on,dummy
1110  data 3,mode change,mono on,dummy
1120  data 3,mode change,poly on,dummy
1130  data 1,undefined
1140  data 3,song pointer,lsb,msb
1150  data 2,song select,song
1160  data 1,undefined
1170  data 1,undefined
1180  data 1,tune request
1190  data 1,end-of-exclusive
1200  data 1,MIDI clock
1210  data 1,undefined
1220  data 1,start
1230  data 1,continue
1240  data 1,undefined
1250  data 1,stop
1260  data 1,active sensing
1270  data 1,system reset
1280  data 2,system exclusive,manuf.ID
```

Listing 1

the network. Following the SYSEX flag is a single data byte called a Manufacturer I.D. After that comes a sequence of data bytes pertinent to some function of the designated equipment—received and acted upon by it, ignored by everybody else. An End Of Exclusive message marks the end of the data set transmission.

MIDI Analysis To Go

The ST Basic program in Listing 1, Midi Analyst, captures the information presented above in coherent and functional form. Analyst receives and analyzes incoming MIDI data, disassembling the input stream in much the same manner as a sophisticated synthesizer and providing an abbreviated English language translation of each message.

Analyst is designed much like an assembly language program; its "knowledge" of the MIDI protocol is encoded in a set of offset tables, permitting fast response to input by avoiding the unnecessary use of conditional expressions.

Next issue, we'll turn away from theory for a change and check out the cream of the crop of ST MIDI sequencing packages. Until then, keep an eye out for active sensing and avoid being hit in the head by bulk dumps. Ciao. ■

What Next?

After the new wears off, many personal computers wind up gathering dust in a closet. Don't let your Atari be one of them.

Why did you originally buy an Atari computer? To do word processing? To compose music? To manage your business? To play games? Chances are, whatever your initiaf reason for buying an Atari, you've discovered that it has many additional capabilities and potential applications.

The flip side of the coin is that you've probably experienced some frustration as well. Maybe your word processing package won't do subscripts or underlining. Perhaps your database won't sort on as many fields as you need. Or, it could be that when you write a program, your whole system acts user-hostile.

Depending upon the balance between your satisfaction and your frustration, you may continue to use your computer or set it aside. But there is a better way: **Atari Explorer** magazine.

As the premier magazine for Atari computer owners, it is our responsibility to make sure that you get the most out of your computer. To us, that means making sure that your Atari does more than you bought it to do, more than friends and associates' computers do, and, indeed, more than you ever imagined that a computer could do.

To make sure that you get the most out of your computer, **Atari Explorer** brings you objective, in-depth reviews of hardware and software; up-to-date information about new products; practical tutorials; stimulating columns; thought-provoking articles; and valuable inside information.

Hard-hitting Evaluations

At **Atari Explorer**, we obtain new peripherats and software packages as soon as they are released. We put them through their paces in our on-site faboratory and also in the environment tor which they are intended: home, office, lab, or school.

Our evaluations are unbiased and accurale. We are not atraid to call a spade a spade or a lemon a lemon. Our first obligation is to you, our readers, and editorial excellence and integrity are our highest goals.

Practical and Stimulating

We know that some of our readers are beginners and others are experts. Thus, it is our responsibility to make whal we publish both comprehensible to newcomers and interesting to veterans. That does not necessarily mean that the material is simple; we know you like to be challenged. What it does mean is that we provide the inexperienced user with every possibfe means to seize the subject matter and make it his own.

However, we don't want the experts to be bored, so although articles are accessible to beginners, they are theoretically non-trivial, cover topics in depth, and present information on more than one level.

At **Atari Explorer**, we are intensely interested in all aspects of computing. Ours is the magazine of pragmatic applications, communicative graphics, stunning animation, mind-expanding games, and realistic simulations. We take our business seriously, but we have fun loo. We are convinced that you will, too, when you go exploring with the **Atari Explorer** family.

A Great Deal! Guaranteed!

To order your subscription to **Atari Explorer**, send a check or money order for $14.95 for six Issues or $39.95 for 18 issues (25% saving over the newsstand price) to: **Atari Explorer**, 7 Hilltop Road, Mendham, NJ 07945. If you preter, you may charge your subscription to your Visa or Master Card. We guarantee your complete satisfaction or we will refund the unfulfilled portion of your subscription.

Atari Explorer
7 Hilltop Road
Mendham, NJ 07945

Yes, enter my subscription to **Atari Explorer.**
☐ 6 issues, $14.95*
☐ 18 issues, $39.95*

* Canadian subscribers, add $5.00 per 6 issues; overseas subscribers, add $10.00 per 6 issues. Checks must be in U.S. funds and drawn on a U.S. bank.

Name _____

Company _____

Address _____

City _____ State ____ Zip ____

☐ Payment enclosed ☐ MasterCard ☐ Visa

Credit card number _____

Expiration date _____

Signature _____

Compatibility.

It's All In The Connection.

com·pat·i·bil·i·ty (kəm pat'ə bil'i tē) *n.* 1. capable of living together harmoniously or getting along well together.

Sounds more like the kind of relationship shared between best friends rather than between Atari 8-bit computers and non-Atari peripherals, *doesn't it?*

Not anymore.

Not since we've defined peripheral compatibility for the Atari XL/XE systems.

It's about time you were able to choose a printer or MODEM because it possessed the special features needed to enhance your system's performance. *Right?*

And isn't it about time your selection was not just limited to only a handful of peripherals boasting 'Atari Compatibility' as their most impactful benefit?

Now, the time has come to change your peripheral habits . . . *for the better.* Our compatible connections give you choices you've never had before, bridging the gap between the hundreds of printers and MODEMs available today and the Atari 8-bit world.

The P:R:Connection is packed full of surprises . . . a standard 'centronics' parallel printer port plus, not one but two serial/MODEM ports! The serial ports resemble that of the Atari 850 interface, possessing all the same signals and functions while using a fully compatible R:handler. Simply plug it into the serial port of your Atari 8-bit and suddenly the power of compatibility enables you to explore new and exciting dimensions in today's printer and MODEM resources.

Our Printer Connection does just what its name implies . . . supplies a compatible connection between your Atari 8-bit and the standard 'centronics' parallel printer of your choice!

Either way you look at it, both connections have clearly defined *"compatibility"* for the Atari 8-bit user. Finally, peripheral choice is yours at last. Now, only the hard part is left up to you . . . *making up your mind!*

For further ICD product information, please call or write for our catalog today.

1220 Rock Street
Rockford, IL 61101-1437
(815) 968-2228
MODEM: (815) 968-2229
FAX: (815) 968-6888